

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Kebutuhan Sistem

Untuk implementasi sistem ini ada beberapa spesifikasi perangkat lunak dan perangkat keras yang dibutuhkan.

4.1.1 Kebutuhan Perangkat Keras

Perangkat keras adalah komponen fisik peralatan yang membentuk sistem komputer, serta peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Untuk menjalankan aplikasi ini membutuhkan komputer dengan spesifikasi minimum sebagai berikut :

1. *Processor 1Ghz Dual Core*
2. Kapasitas *Random Access Memory (RAM) 2GB*
3. *Hard disk dengan free space 10GB*
4. *VGA on Board*
5. *Monitor dengan resolusi 1366x768*
6. *Keyboard dan mouse*

4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak merupakan komponen non fisik yang digunakan untuk membuat sistem komputer dapat berjalan dan melakukan tugasnya. Adapun perangkat lunak yang dibutuhkan adalah sebagai berikut :

1. Kebutuhan *Minimum Client*

Adapun perangkat lunak yang dibutuhkan dan telah diujicobakan pada komputer *client* yaitu:

- a. *Operating System : Windows XP Service Pack 2*
 - b. *Browser : Mozilla Firefox versi 12.0*
2. *Kebutuhan Minimum Server*

Adapun perangkat lunak yang dibutuhkan dan telah diujicobakan pada komputer *server* yaitu:

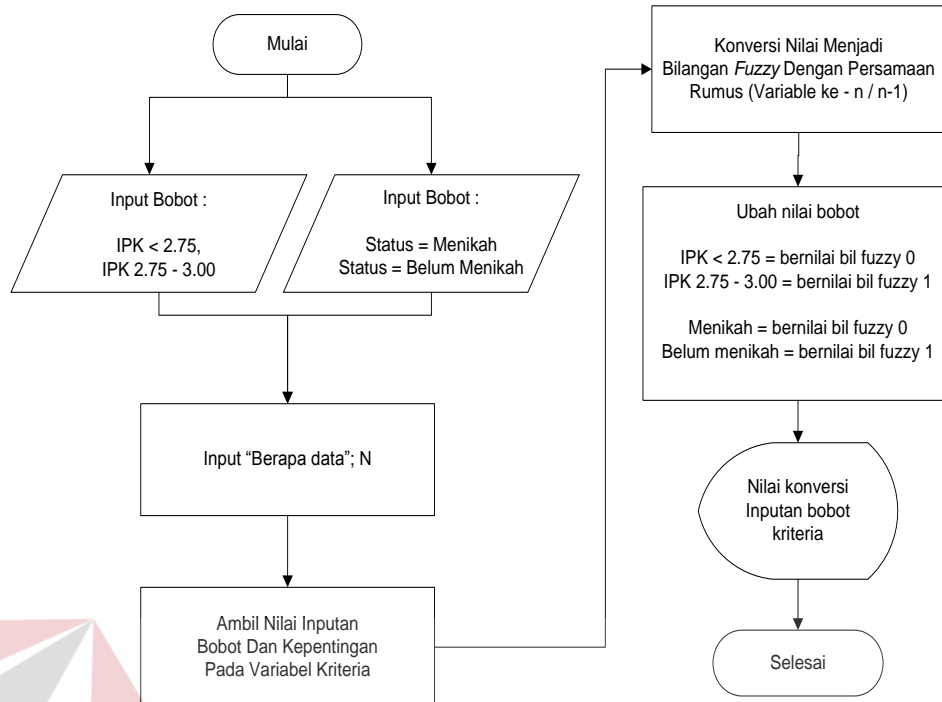
- a. *Operating System : Windows XP Service Pack 2*
- b. *Web Server : Apache 2.4.3*
- c. *Programming Language : PHP 5.4.7*
- d. *Database : MySQL 5.5.27*

4.2 Implementasi Sistem

Implementasi adalah realisasi dari sebuah aplikasi, pelaksanaan dari suatu rencana, ide, model, algoritma atau kebijakan. Didalam ilmu komputer, implementasi merupakan realisasi dari spesifikasi teknis atau algoritma sebagai sebuah program, komponen perangkat lunak atau sistem komputer lainnya melalui pemrograman dan pengembangan. Berikut adalah penjelasan dari proses-proses yang ada pada aplikasi berdasarkan rancangan yang dibuat pada bab tiga.

4.2.1 Implementasi Penerapan Merubah Bilangan *Fuzzy*

Pada proses ini dilakukan untuk merubah bobot kriteria seleksi menjadi nilai bilangan *fuzzy* yang akan digunakan untuk melakukan proses perhitungan normalisasi. Berikut merupakan *flowchart* konversi bilangan *fuzzy*.



Gambar 4.1 *Flowchart Konversi Bilangan Fuzzy*

```

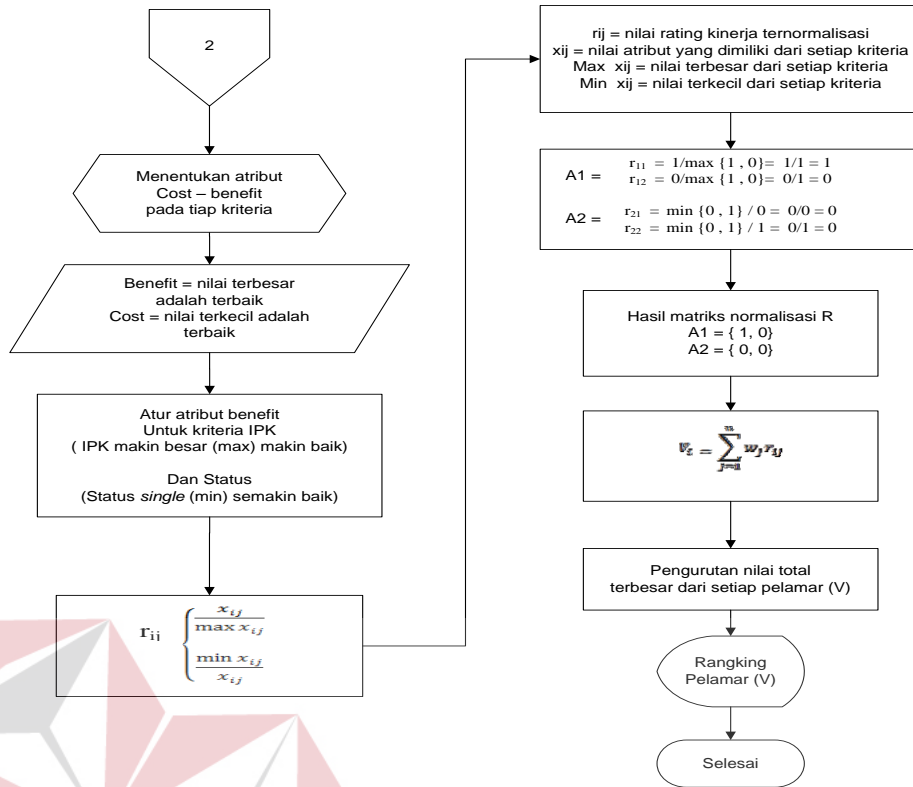
    $jml = mysql_query("select count(*) as jml from RangeBobot where JenisBobot = '". $ _POST['txtJenis']."' and email = '". $ _SESSION['emailPT']."'");
    $row1 = mysql_fetch_array($jml);

    $q = mysql_query("select * from RangeBobot where JenisBobot = '". $ _POST['txtJenis']."' and email = '". $ _SESSION['emailPT']."'
    order by urut;");
    while ($row = mysql_fetch_array($q)) {
        $nilai = ($row['urut'] / ($row1['jml'] - 1));
        $update = mysql_query("update RangeBobot set Nilai = '". $ nilai."' where urut = '". $ row['urut']."' and JenisBobot = '". $ _POST['txtJenis']."' and email = '". $ _SESSION['emailPT']."'");
    }
  
```

Gambar 4.2 *Source Code Konversi Bilangan Fuzzy*

4.2.2 Implementasi Perhitungan Normalisasi

Pada proses ini perhitungan matriks normalisasi kriteria seleksi sampai dengan menghasilkan nilai tertinggi yang terpilih sebagai alternatif terbaik.



Gambar 4.3 Flowchart Perhitungan Matriks Normalisasi (R)

```

$gKriteria = mysql_query("select email, emailPT, idLamaran, usia, pend, akreditasi, ipk, pengalaman, tinggi, berat, menikah, kacamata
from kriteria where emailPT = ".$emailPT." and idLamaran = ".$idLamaran.");
while ($r = mysql_fetch_array($gKriteria)){
    $hasilUsia = $usia_min / $r['usia'];
    $hasilPend = $r['pend'] / $pend_max;
    $hasilAkre = $r['akreditasi'] / $akre_max;
    $hasilIPK = $r['ipk'] / $ipk_max;
    $hasilPengalaman = $r['pengalaman'] / $peng_max;
    $hasilTinggi = $r['tinggi'] / $peng_tinggi;
    $hasilBerat = $berat_min / $r['berat'];
    $hasilMenikah = $r['menikah'] / $peng_meni;
    $hasilKc = $r['kacamata'] / $peng_kc;

```

Gambar 4.4 Source Code Menentukan Atribut Cost Dan Benefit

```

$sql = mysql_query("select nor.email as email, concat(d>NamaDepan, ' ', d>NamaBelakang) as Nama,
round((kep.usia * nor.usia) + (kep.pendidikan * nor.pend) + (kep.akreditasi * nor.akreditasi) +
(kep.ipk * nor.ipk) + (kep.pengalaman * nor.pengalaman) + (kep.tinggi * nor.tinggi) + (kep.berat * nor.berat)
+ (kep.menikah * nor.menikah) + (kep.kacamata * nor.kacamata),2) as Rangking

from normalisasi nor, kepentingan kep, detilpribadi d, lamaran l
where nor.emailPT = kep.email and nor.email = d.Email and l.id = nor.idLamaran and l.email = nor.emailPT and
nor.emailPT = ".$emailPT." and nor.idLamaran = ".$idLamaran."
order by Rangking desc");

```

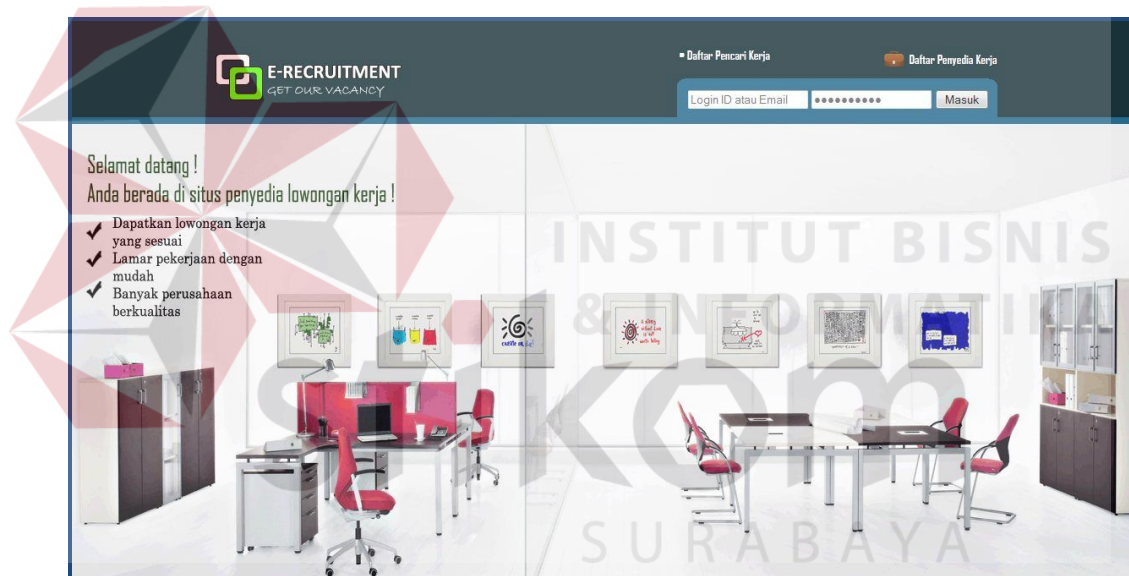
Gambar 4.5 Source Code Perhitungan Rangking

4.2.3 Implementasi Antarmuka

Berikut adalah beberapa antarmuka yang telah diimplementasikan sesuai desain antarmuka pada bab tiga.

A. Halaman Utama

Gambar 4.6 merupakan halaman utama saat aplikasi dari sistem informasi seleksi administrasi *online* ini dijalankan. Pada halaman ini akan terdapat beberapa menu yaitu meliputi menu daftar pencari kerja, daftar penyedia kerja, *input login*, *input* kata sandi dan tombol masuk.

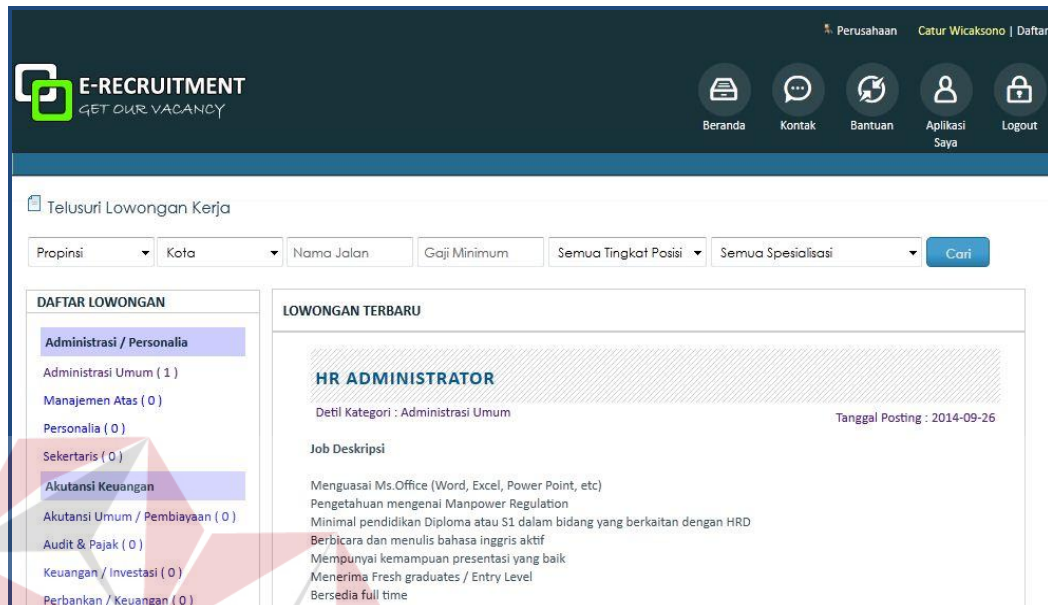


Gambar 4.6. Halaman Utama Aplikasi

B. Halaman Utama Pencari Kerja

Gambar 4.7 merupakan halaman utama pencari kerja, merupakan tampilan awal saat *user* pencari kerja berhasil *login*. Pada halaman ini akan terdapat beberapa menu untuk melakukan proses selanjutnya sesuai dengan hak aksesnya, menu bagian atas meliputi menu beranda, kontak, bantuan, aplikasi saya dan

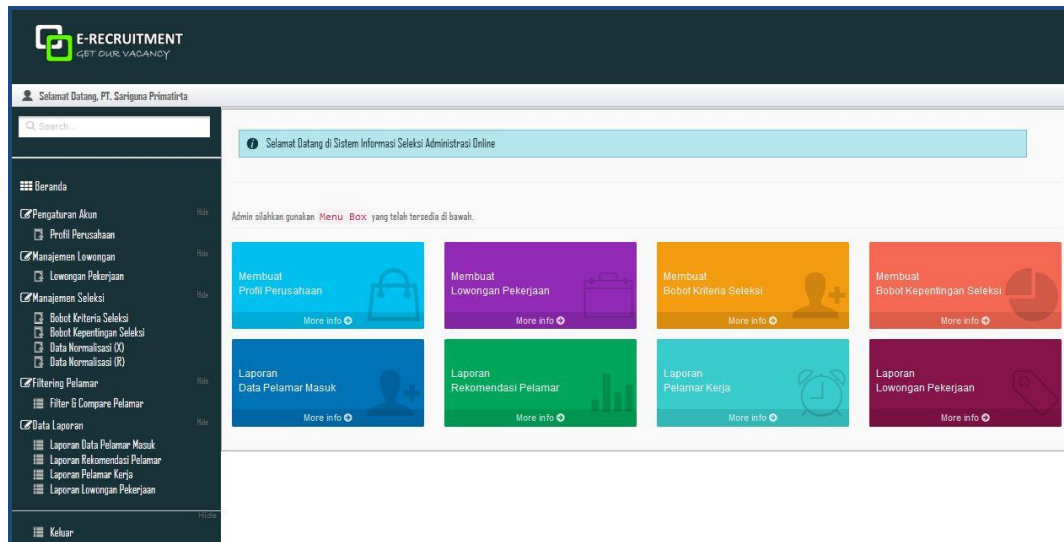
logout. Sedangkan menu bagian bawah meliputi menu telusuri lowongan, daftar lowongan kerja dan menu informasi lowongan kerja terbaru.



Gambar 4.7. Halaman Utama Pencari Kerja

C. Halaman Utama Penyedia Kerja

Gambar 4.8 merupakan halaman utama pencari kerja, merupakan tampilan awal saat *user* penyedia kerja berhasil *login*. Pada halaman ini akan terdapat beberapa menu yang dapat dikelola ataupun menu yang menyajikan informasi bagi *user* penyedia kerja, yaitu menu pengaturan akun, manajemen lowongan, manajemen seleksi, *filtering* pencari kerja dan data laporan.



Gambar 4.8. Halaman Utama Penyedia Kerja

D. Halaman *Input* Pendaftaran Penyedia Kerja

Gambar 4.9 merupakan halaman pendaftaran penyedia kerja, merupakan halaman yang digunakan untuk memasukkan data-data untuk keperluan pendaftaran pengguna baru dalam hal ini bagi *user* penyedia kerja.

Negara

Propinsi / Daerah

Kota

Detil Alamat

Spesialisasi Bidang Usaha

Gambar 4.9. Halaman *Input* Pendaftaran Penyedia Kerja

E. Halaman *Input Login* Penyedia Kerja

Gambar 4.10 merupakan halaman *login user* penyedia kerja, merupakan halaman yang digunakan *user* penyedia kerja agar dapat masuk ke halaman utama administrasi penyedia kerja.



Gambar 4.10. Halaman *Input Login* Penyedia Kerja

F. Halaman *Input Pendaftaran* Pencari Kerja

Gambar 4.11 merupakan halaman pendaftaran pencari kerja, merupakan halaman yang digunakan untuk memasukkan data-data keperluan pendaftaran pengguna baru dalam hal ini untuk *user* pencari kerja.



Gambar 4.11. Halaman *Input Pendaftaran* Pencari Kerja

G. Halaman *Input Login* Pencari Kerja

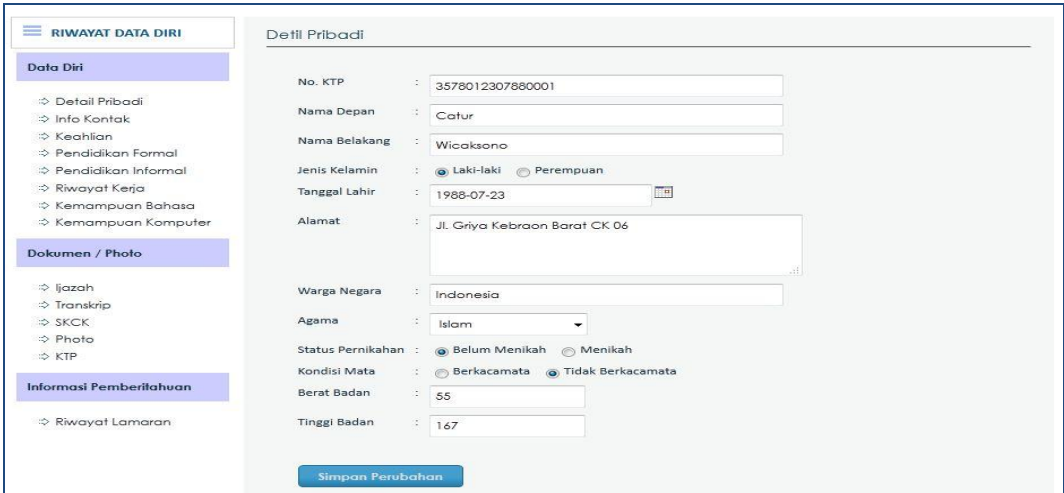
Gambar 4.12 merupakan halaman *login* pencari kerja, merupakan halaman yang digunakan *user* pencari kerja untuk dapat masuk ke halaman utama aplikasi dengan memasukkan *login id* atau *username* dan kata sandi atau *password*.



Gambar 4.12. Halaman *Input Login* Pencari Kerja

H. Halaman *Input Data* Pencari Kerja

Gambar 4.13 merupakan halaman *input* data pencari kerja, merupakan halaman tempat mengelola data profil pencari kerja, seperti mengelola data detail pribadi, info kontak, pendidikan formal dan non formal, riwayat kerja dan data lainnya.



Gambar 4.13. Halaman *Input Data* Pencari Kerja

I. Halaman *Input* Pencarian Lowongan Kerja

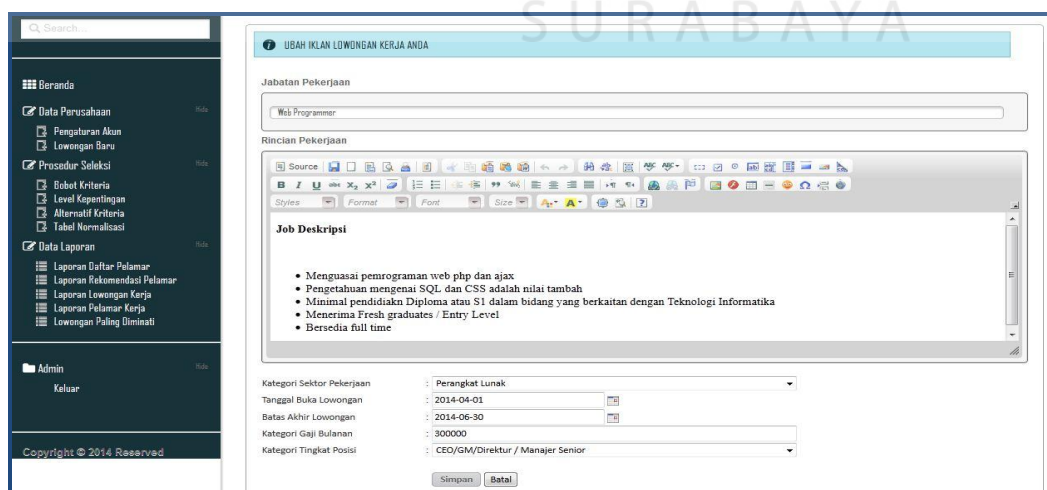
Gambar 4.14 merupakan halaman *input* pencarian lowongan kerja, merupakan halaman untuk melakukan pencarian lowongan kerja. Pada halaman pencarian lowongan ini terdapat beberapa kategori yang dapat digunakan sebagai indikator pencarian tersebut antara lain kategori propinsi, kota dan gaji minimum.



Gambar 4.14. Halaman *Input* Pencarian Lowongan Kerja

J. Halaman *Input* Lowongan Kerja

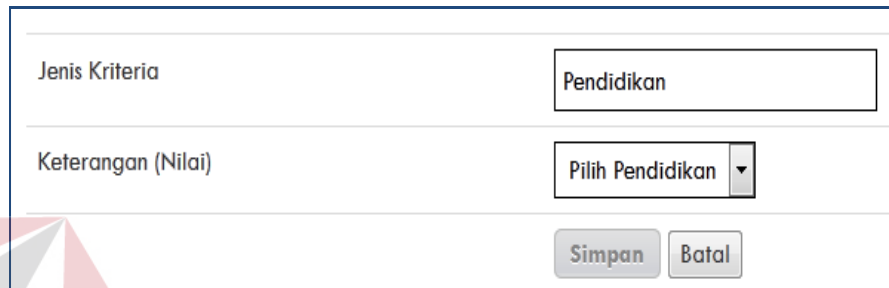
Gambar 4.15 merupakan halaman *input* data lowongan kerja, merupakan halaman tempat mengelola data-data untuk keperluan pembukaan lowongan kerja antara lain kategori sektor pekerjaan, tanggal pembukaan dan penutupan lowongan kerja dan kategori lain.



Gambar 4.15. Halaman *Input* Lowongan Kerja

K. Halaman *Input* Bobot Kriteria

Gambar 4.16 merupakan halaman *input* data bobot kriteria, merupakan halaman tempat mengelola nilai bobot kriteria yang digunakan sebagai variabel-variabel untuk proses perhitungan ranking sehingga menghasilkan rekomendasi pencari kerja.

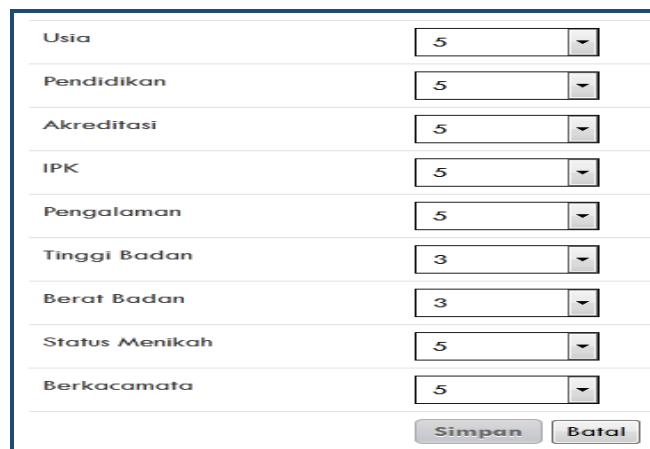


Jenis Kriteria	Pendidikan
Keterangan (Nilai)	Pilih Pendidikan ▼
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Gambar 4.16. Halaman *Input* Bobot Kriteria

L. Halaman *Input* Bobot Kepentingan

Gambar 4.17 merupakan halaman *input* data bobot kepentingan, sama seperti pada Gambar 4.16 merupakan halaman tempat mengelola nilai bobot kepentingan yang digunakan sebagai variabel-variabel untuk proses penyeleksian pencari kerja.



Usia	5 ▼
Pendidikan	5 ▼
Akreditasi	5 ▼
IPK	5 ▼
Pengalaman	5 ▼
Tinggi Badan	3 ▼
Berat Badan	3 ▼
Status Menikah	5 ▼
Berkacamata	5 ▼
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Gambar 4.17. Halaman *Input* Bobot Kepentingan

M. Halaman *Input Filtering* Pencari Kerja

Gambar 4.18 merupakan halaman *input filtering* data pencari kerja, merupakan halaman untuk menyaring pencari kerja dengan pilihan kategori yang dapat dipilih. Untuk menyaring pencari kerja berdasarkan kategori usia, *user* penyedia kerja dapat memasukkan data pada kolom usia dengan indikator inputan *Min* dan *Max*. Jika ingin menyaring berdasarkan kategori lain dapat memilih kemudian menekan tombol cari.

Gambar 4.18. Halaman *Input Filtering* Pencari Kerja

N. Halaman *Output* Matriks Keputusan

Gambar 4.19 merupakan halaman *output* matriks keputusan. Di halaman ini *user* penyedia kerja dapat melihat informasi nilai masing-masing kriteria seleksi dari pencari kerja yang telah dikonversi menjadi bilangan *fuzzy*.

Email Pelamar	Nama Pelamar	Nilai Matriks Keputusan Kriteria Seleksi (X)								
		Usia	Pendidikan	Akreditasi	IPK	Pengalaman	Tinggi	Berat	Status	Kacamata
agus@gmail.com	Agus Hafiludin	0.67	0.57	1.00	0.40	0.60	1.00	0.20	0.00	0.00
atun@gmail.com	Atun Istiqomah	1.00	0.71	1.00	0.60	0.40	0.40	1.00	0.00	1.00
catur@gmail.com	Catur Wicaksono	0.67	0.71	0.67	0.60	0.20	0.80	0.80	1.00	1.00
sri@gmail.com	Sri Wahyuningsih	1.00	0.71	1.00	0.60	0.40	0.40	1.00	0.00	1.00
supri@yahoo.com	Supriyadi Asmanto	0.67	0.57	1.00	1.00	1.00	0.60	0.60	1.00	1.00

Gambar 4.19. Halaman *Output* Matriks Keputusan

O. Halaman *Output* Matriks Normalisasi

Gambar 4.20 merupakan halaman *output* matriks normalisasi. Di halaman ini penyedia kerja dapat melihat hasil dari perhitungan matriks normalisasi berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan/*benefit* ataupun atribut biaya/*cost*) terhadap data pencari kerja yang telah diinputkan kemudian ditampilkan ke dalam sebuah laporan yang berisi besaran nilai masing-masing kriteria seleksi.

Email Pelamar	Nama Pelamar	Nilai Matriks Normalisasi Kriteria Seleksi (R)								
		Usia	Pendidikan	Akreditasi	IPK	Pengalaman	Tinggi	Berat	Status	Kacamata
agus@gmail.com	Agus Hafiludin	1.00	0.66	1.00	0.40	0.60	1.00	1.00	0.00	0.00
atun@gmail.com	Atun Istiqomah	0.67	0.83	1.00	0.60	0.40	0.40	0.20	0.00	0.00
catur@gmail.com	Catur Wicaksono	1.00	0.83	0.67	0.60	0.20	0.80	0.25	0.00	0.00
sri@gmail.com	Sri Wahyuningsih	0.67	0.83	1.00	0.60	0.40	0.40	0.20	0.00	0.00
supri@yahoo.com	Supriyadi Asmanto	1.00	0.66	1.00	1.00	1.00	0.60	0.33	0.00	0.00

Gambar 4.20. Halaman *Output* Matriks Normalisasi

P. Halaman *Output* Rekomendasi Pencari Kerja

Gambar 4.21 merupakan halaman *output* rekomendasi pencari kerja, merupakan halaman yang menampilkan informasi mengenai rekomendasi pencari kerja berdasarkan perhitungan sistem. Pada halaman ini ditampilkan lima rekomendasi pencari kerja dengan nilai tertinggi. Dengan demikian didapatkan informasi mengenai pencari kerja yang sesuai dengan kriteria dan kebutuhan perusahaan, yang kemudian akan dipilih sebagai alternatif terbaik.

Rekomendasi Pelamar				
Lowongan Pekerjaan: Marketing				
Peringkat	Nama Kandidat	Keterangan	Nilai (poin)	Aksi
1	Yanuar Wibowo ✉ Email : yanuar@gmail.com	Usia : 29 Tahun Pendidikan : Magister - S2 Akreditasi : A IPK / NEM : 3.8 Pengalaman : 5 Tahun Berkacamata : Tidak Status : Menikah Tinggi Badan : 169 Cm Berat Badan : 75 Kg	5.90	Update Status Compare
2	Supriyadi Asmanto ✉ Email : supri@yahoo.com	Usia : 28 Tahun Pendidikan : Diploma - D3 Akreditasi : A IPK / NEM : 3.8 Pengalaman : 5 Tahun Berkacamata : Tidak Status : Belum Menikah Tinggi Badan : 165 Cm Berat Badan : 57 Kg	5.14	Update Status Compare

Gambar 4.21. Halaman *Output* Rekomendasi Pencari Kerja

Q. Halaman *Output Compare* Pencari Kerja

Gambar 4.22 merupakan halaman *output compare* data pencari kerja, pada halaman ini disajikan perbandingan antara dua data pencari kerja. Selain itu ditampilkan pula informasi nilai perhitungan untuk mendukung spesifikasi perbandingan kedua pencari kerja tersebut.

Head To Head		
Pelamar 1	VS	Pelamar 2
Marketing	Judul Lowongan	Civil And Structural Engineer
Yanuar Wibowo	Nama	Catur Wicaksono
Laki-laki	Jenis Kelamin	Laki-laki
1986-02-17	Kelahiran	1988-07-23
View Details		View Details

Gambar 4.22. Halaman *Output Compare* Pencari Kerja

R. Halaman *Output* Pencari Kerja Masuk

Gambar 4.23 merupakan halaman *output* pencari kerja masuk, merupakan halaman berisi tentang laporan pencari kerja yang telah melamar pada perusahaan

tersebut. Pada halaman ini penyedia kerja dapat melihat detail pencari kerja dengan menekan tombol *view detail*.

Berikut ini daftar pelamar yang ada :

No	Judul Lowongan	Email Pelamar	Nama Pelamar	Detil
1	Marketing	sri@gmail.com	Sri Wahyuningsih	View Details
2	Marketing	agus@gmail.com	Agus Hafiludin	View Details
3	Marketing	atun@gmail.com	Atun Istiqomah	View Details
4	Marketing	yanuar@gmail.com	Yanuar Wibowo	View Details
5	Marketing	supri@yahoo.com	Supriyadi Asmanto	View Details

Halaman : 1 | 2 | 3 |

Gambar 4.23. Halaman *Output* Pencari Kerja Masuk

S. Halaman *Output* Detail Pencari Kerja

Gambar 4.24 merupakan halaman *output* detail pencari kerja, merupakan halaman yang digunakan sebagai pelengkap dari halaman *output* pencari kerja masuk, dikarenakan pada *output* pencari kerja masuk tidak diberikan informasi dari para pencari kerja secara lengkap. Pada halaman ini *user* penyedia kerja dapat mencetak data pencari kerja tersebut dan menjadikannya sebagai informasi *curriculum vitae* pencari kerja.

Catur Wicaksono
 Jl. Griya Kebraon Barat CK No.6 Surabaya Jawa timur Indonesia
 Email : catur@gmail.com, Tel : - (Rumah) , 085731027873 (Hp)
 -

Ringkasan Resume

Pendidikan Formal	Pendidikan Informal	Riwayat Kerja	Kemampuan Bahasa	Kemampuan Komputer	Dokumen
Pendidikan	:	Sarjana - S1			
Institusi	:	STIKOM Surabaya			
Fakultas	:	Sistem Informasi			
Negara	:	Indonesia			
Tahun Masuk	:	2007			
Tahun Lulus	:	2011			
Nilai IPK / NEM	:	3.5			

Gambar 4.24. Halaman *Output* Detail Pencari Kerja

T. Halaman *Output* Pencari Kerja

Gambar 4.25 dan Gambar 4.26 merupakan halaman *output* pencari kerja, berisi tentang informasi pencari kerja yang telah melamar di dalam penyedia kerja tersebut. Pada halaman *output* pencari kerja terdapat dua laporan yaitu laporan pencari kerja per periode dan laporan pencari kerja per tingkat posisi.

Periode :

s/d

Hasil Pencarian Data Yang Ditemukan Untuk Periode (2015-01-01 s/d 2015-03-01) Sebanyak = 12 Pelamar

Email Pelamar	Nama Pelamar	Judul Lowongan	Tgl Lamar	Status Lamar
yanuar@gmail.com	Yanuar Wibowo	Marketing	2015-02-26	Tidak Ada Status
catur@gmail.com	Catur Wicaksono	Marketing	2015-02-23	Tidak Ada Status

Gambar 4.25. Halaman *Output* Pencari Kerja Kerja Per Periode

Tingkat Posisi :

Pilih Tingkat Posisi

Hasil Pencarian Data Yang Ditemukan Untuk Posisi (Staf Administrasi) Sebanyak = 6 Pelamar

Email Pelamar	Nama Pelamar	Judul Lowongan	Tingkat Posisi	Tgl Lamar	Status Lamar
sri@gmail.com	Sri Wahyuningsih	Marketing	Staf Administrasi	2015-02-11	Tidak Ada Status
agus@gmail.com	Agus Hafiludin	Marketing	Staf Administrasi	2015-02-11	Telah Diterima

Gambar 4.26. Halaman *Output* Pencari Kerja Kerja Per Tingkat Posisi

U. Halaman *Output* Lowongan Kerja

Gambar 4.27, Gambar 4.28 dan Gambar 4.29 merupakan halaman *output* lowongan kerja, berisi tentang informasi lowongan kerja pernah perusahaan tersebut tawarkan. Pada halaman *output* lowongan kerja terdapat tiga laporan

yaitu laporan lowongan kerja per periode, laporan lowongan kerja per tingkat posisi dan lowongan kerja paling diminati.

Periode :

s/d

Hasil Pencarian Data Yang Ditemukan Untuk Periode (2015-01-01 s/d 2015-03-01) Sebanyak = 2 Lowongan

No	Judul Lowongan	Tgl Posting
1	Marketing	2015-02-26
2	Civil And Structural Engineer	2015-02-05

Gambar 4.27. Halaman *Output* Lowongan Kerja Per Periode

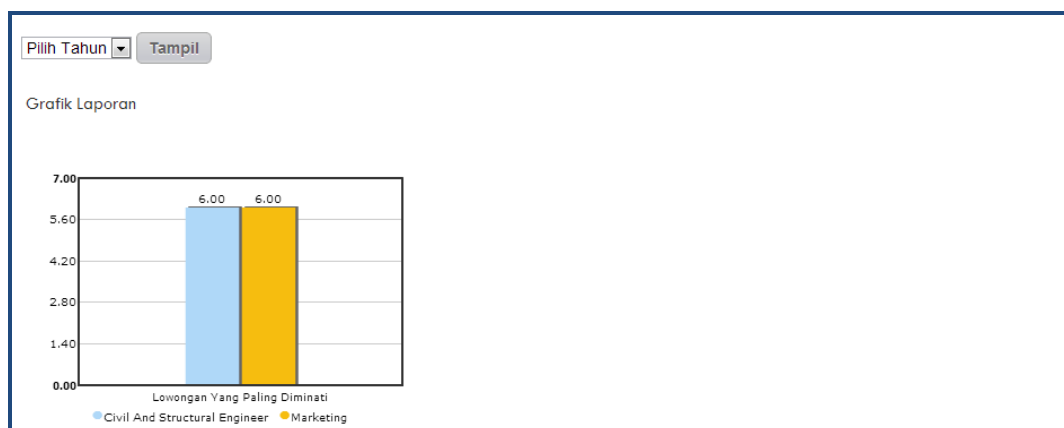
Tingkat Posisi :

Pilih Tingkat Posisi

Hasil Pencarian Data Yang Ditemukan Untuk Posisi (General Affair / Bagian Umum) Sebanyak = 1 Lowongan

No	Judul Lowongan	Tingkat Posisi
1	Civil And Structural Engineer	General Affair / Bagian Umum

Gambar 4.28. Halaman *Output* Lowongan Kerja Per Tingkat Posisi



Gambar 4.29. Halaman *Output* Lowongan Kerja Paling Diminati

4.3 Evaluasi Dan Pembahasan

4.3.1 Evaluasi Sistem

Evaluasi dilakukan untuk melakukan pengujian sistem, apakah sistem yang telah dibuat dapat berjalan sesuai dengan tujuan. Jika terjadi perbedaan hasil maka sistem yang dibuat masih memiliki kesalahan. Proses pengujian menggunakan *Black Box Testing* dimana aplikasi diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang dibuat telah sesuai dengan tujuan. Berikut akan dijelaskan hasil uji coba yang telah dilakukan antara lain:

1. Uji coba fungsi aplikasi.
2. Uji coba penghitungan sistem menggunakan metode *Simple Additive Weighting* (SAW).

A. Uji Coba Fungsi Aplikasi

Pada tahapan ini adalah tahapan uji coba sistem yang akan digunakan. Penulis melakukan uji coba dengan mengacu pada fitur utama yang terdapat pada aplikasi *web* utama dengan cara menempatkan pada tabel dan memberikan *input* ke dalam fitur yang akan di tes.

A.1. Uji Coba Fungsi Fitur Utama Sistem

A.1.1. Uji Coba *Form* Pendaftaran Penyedia Kerja

Pada *form* pendaftaran penyedia kerja ini terdapat proses untuk menyimpan data *user* penyedia kerja, dipakai untuk bisa masuk kedalam aplikasi sistem informasi seleksi administrasi *online*, jika ada *user* penyedia kerja yang belum terdaftar, maka *user* penyedia kerja tersebut tidak dapat mengakses program sistem informasi seleksi administrasi *online* ini.

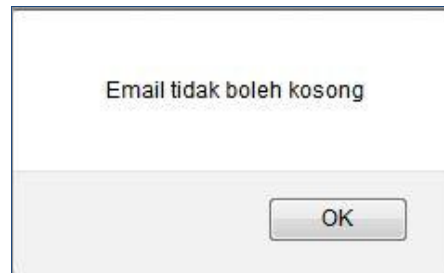
Pada uji coba *form* pendaftaran penyedia kerja ada beberapa uji coba yang dilakukan. Uji coba pertama untuk menguji kelengkapan isian data dari *user*. Selanjutnya uji coba untuk menghindari penggunaan data yang sama. Uji coba yang terakhir menguji proses simpan data ketika masukan telah benar dan lengkap. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan menggunakan tabel uji coba pada Tabel 4.1.

Tabel 4.1 Uji Coba *Form* Pendaftaran Penyedia Kerja

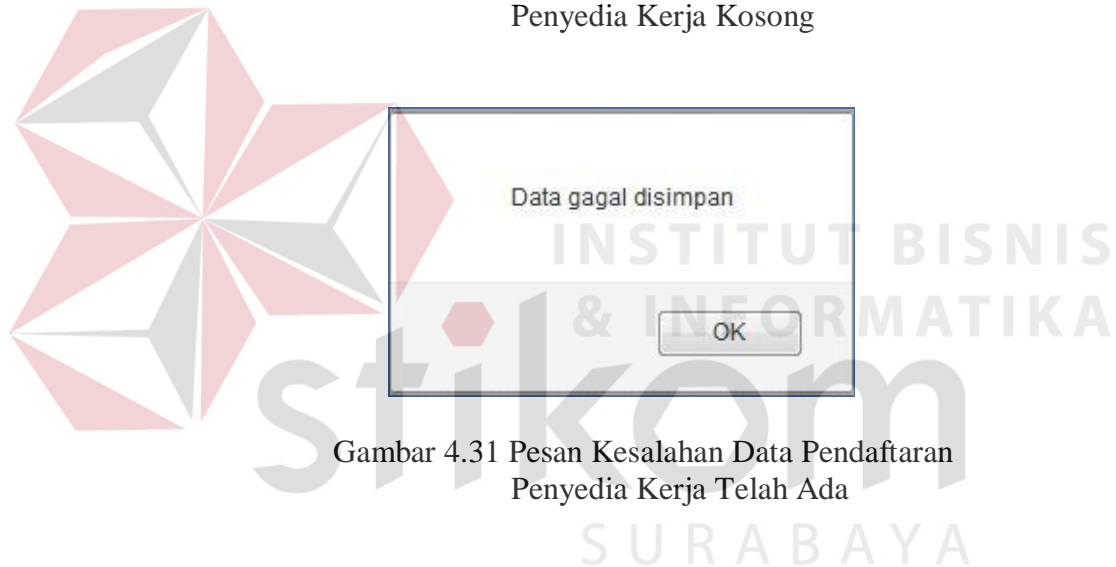
Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
1	Menguji respon sistem ketika kolom inputan kosong	Salah satu kolom inputan data pendaftaran tidak diisi	Muncul pesan kesalahan “Inputan Tidak Boleh Kosong” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran tidak berhasil 3. Muncul pesan kesalahan “Isian Tidak Boleh Kosong”
2	Menguji respon sistem ketika <i>email id</i> penyedia kerja sama	Memasukkan data <i>email id</i> sama dengan <i>email id</i> yang telah ada	Muncul pesan kesalahan “Data Gagal Disimpan” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran tidak berhasil 3. Muncul pesan kesalahan “Data Gagal Disimpan”
3	Menguji respon sistem ketika kolom inputan diisi dengan benar	Memasukkan data pendaftaran dengan benar dan lengkap	Muncul pesan konfirmasi “Pendaftaran Berhasil” dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran berhasil 3. Muncul pesan konfirmasi “Pendaftaran Berhasil” 4. Halaman berganti

Guna menghindari kesalahan pengisian data pendaftaran pada *form* pendaftaran penyedia kerja terdapat beberapa validasi yaitu validasi untuk

menghindari dari data pendaftaran yang kosong dan data inputan sama akan ditampilkan pesan kesalahan seperti pada Gambar 4.30 dan Gambar 4.31 yang sesuai dengan tabel uji coba pada Tabel 4.1 nomor 1 dan nomor 2.

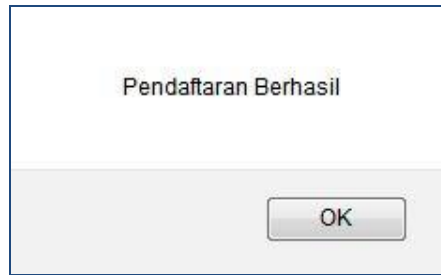


Gambar 4.30 Pesan Kesalahan Data Pendaftaran
Penyedia Kerja Kosong



Gambar 4.31 Pesan Kesalahan Data Pendaftaran
Penyedia Kerja Telah Ada

Sedangkan uji coba pada Tabel 4.1 nomor 3 menghasilkan pesan konfirmasi dari data pendaftaran penyedia kerja, apakah sudah benar dan lengkap atau tidak, dan jika benar dan lengkap akan ditampilkan pesan konfirmasi seperti pada Gambar 4.32.



Gambar 4.32 Pesan Konfirmasi Pendaftaran Penyedia Kerja Berhasil

A.1.2. Uji Coba *Form Login* Penyedia Kerja

Adapun proses hasil uji coba *login* penyedia kerja ini bertujuan untuk mengetahui serta menentukan keberhasilan pada *form login* penyedia kerja. Pada uji coba *login* penyedia kerja terdapat beberapa uji coba yang dilakukan. Uji coba pertama untuk menguji kelengkapan isian data dari *user* penyedia kerja. Selanjutnya uji coba untuk menghindari salah satu data inputan salah. Uji coba yang terakhir menguji proses simpan data ketika masukan telah benar dan lengkap. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan tabel uji coba pada Tabel 4.2.

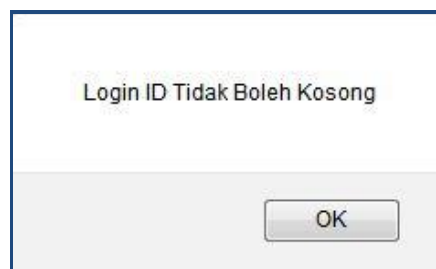
Tabel 4.2 Uji Coba *Form Login* Penyedia Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
4	Menguji respon sistem ketika kolom inputan kosong	Salah satu kolom inputan data <i>login</i> penyedia kerja tidak diisi	Muncul pesan kesalahan "Inputan Tidak Boleh Kosong" dan data tidak tersimpan	1. Sukses 2. <i>Login</i> tidak berhasil 3. Muncul pesan kesalahan "Isian Tidak Boleh Kosong"

Tabel 4.2 Uji Coba *Form Login* Penyedia Kerja (Lanjutan)

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
5	Mengetahui respon sistem ketika salah satu data inputan salah	Salah satu kolom inputan data <i>login</i> penyedia kerja diisi tidak benar	Muncul pesan kesalahan “ <i>Login ID</i> dan Sandi Salah” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. <i>Login</i> tidak berhasil 3. Muncul pesan kesalahan “<i>Login ID</i> dan Sandi Salah”
6	Mengetahui respon sistem ketika kolom inputan diisi dengan benar	Memasukkan data <i>login</i> penyedia kerja dengan benar dan lengkap	Muncul pesan konfirmasi “ <i>Login Berhasil</i> ” dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. <i>Login</i> berhasil 3. Muncul pesan konfirmasi “<i>Login Berhasil</i>” 4. Halaman berganti

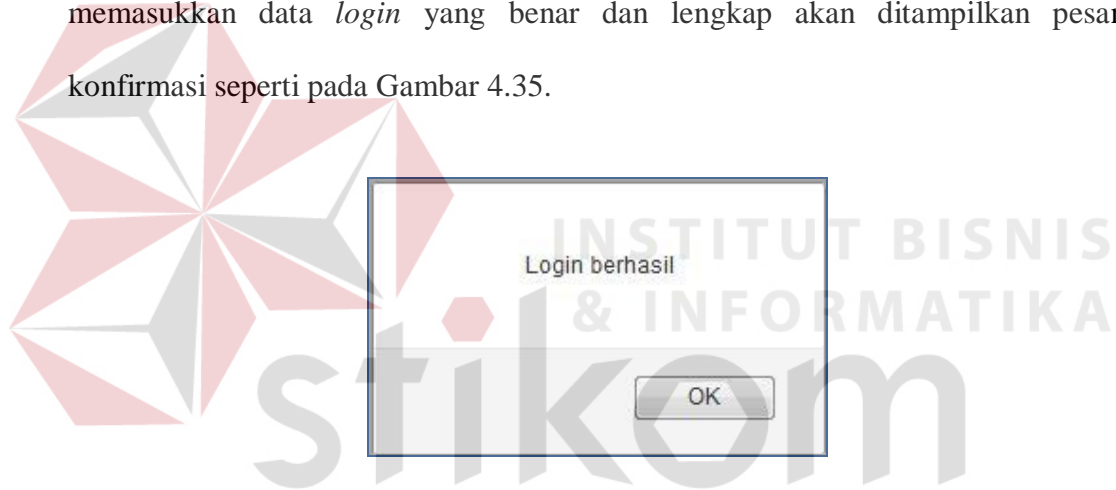
Guna menghindari kesalahan pengisian data *login* pada *form login* penyedia kerja terdapat beberapa validasi yaitu validasi untuk menghindari dari data pendaftaran yang kosong dan salah satu data inputan salah akan ditampilkan pesan kesalahan seperti pada Gambar 4.33 dan Gambar 4.34 yang sesuai dengan tabel uji coba pada Tabel 4.2 nomor 4 dan nomor 5.

Gambar 4.33 Pesan Kesalahan Data *Login* Penyedia Kerja Kosong



Gambar 4.34 Pesan Kesalahan Data *Login* Penyedia Kerja Salah

Sedangkan uji coba pada Tabel 4.2 nomor 6 menghasilkan pesan konfirmasi dari data *login* penyedia kerja, setelah *user* penyedia kerja memasukkan data *login* yang benar dan lengkap akan ditampilkan pesan konfirmasi seperti pada Gambar 4.35.



Gambar 4.35 Pesan Konfirmasi *Login* Penyedia Kerja Berhasil

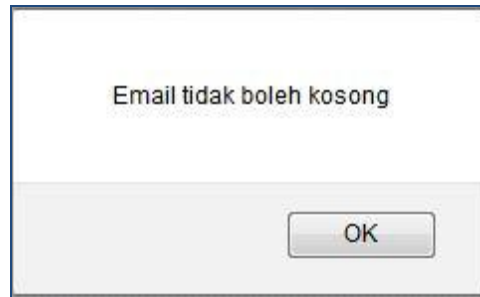
A.1.3. Uji Coba *Form* Pendaftaran Pencari Kerja

Pada uji coba *form* pendaftaran pencari kerja ada beberapa uji coba yang dilakukan. Uji coba pertama untuk menguji kelengkapan isian data dari *user*. Selanjutnya uji coba untuk menghindari penggunaan data yang sama. Uji coba yang terakhir menguji proses simpan data ketika masukan telah benar dan lengkap. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan menggunakan tabel uji coba pada Tabel 4.3.

Tabel 4.3 Uji Coba *Form* Pendaftaran Pencari Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
7	Menguji respon sistem ketika kolom inputan kosong	Salah satu kolom inputan data pendaftaran tidak diisi	Muncul pesan kesalahan “Inputan Tidak Boleh Kosong” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran tidak berhasil 3. Muncul pesan kesalahan “Isian Tidak Boleh Kosong”
8	Menguji respon sistem ketika <i>email id</i> pencari kerja telah ada	Memasukkan data <i>email id</i> sama dengan <i>email id</i> yang telah ada	Muncul pesan kesalahan “Data Gagal Disimpan” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran tidak berhasil 3. Muncul pesan kesalahan “Data Gagal Disimpan”
9	Menguji respon sistem ketika kolom inputan diisi dengan benar	Memasukkan data pendaftaran dengan benar dan lengkap	Muncul pesan konfirmasi “Pendaftaran Berhasil” dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Pendaftaran berhasil 3. Muncul pesan konfirmasi “Pendaftaran Berhasil” 4. Halaman berganti

Guna menghindari kesalahan pengisian data pendaftaran pada *form* pendaftaran pencari kerja terdapat beberapa validasi yaitu validasi untuk menghindari dari data pendaftaran yang kosong dan data inputan sama akan ditampilkan pesan kesalahan seperti pada Gambar 4.36 dan Gambar 4.37 yang sesuai dengan tabel uji coba pada Tabel 4.3 nomor 7 dan nomor 8.

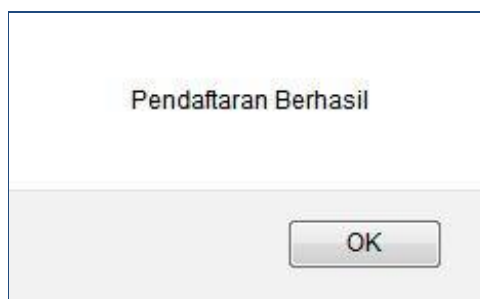


Gambar 4.36 Pesan Kesalahan Data Pendaftaran Pencari Kerja Kosong



Gambar 4.37 Pesan Kesalahan Data Pendaftaran Pencari Kerja Telah Ada

Sedangkan uji coba pada Tabel 4.3 nomor 9 menghasilkan pesan konfirmasi dari data pendaftaran pencari kerja, apakah sudah benar dan lengkap atau tidak, dan jika benar dan lengkap akan ditampilkan pesan konfirmasi seperti pada Gambar 4.38.



Gambar 4.38 Pesan Konfirmasi Pendaftaran Pencari Kerja Berhasil

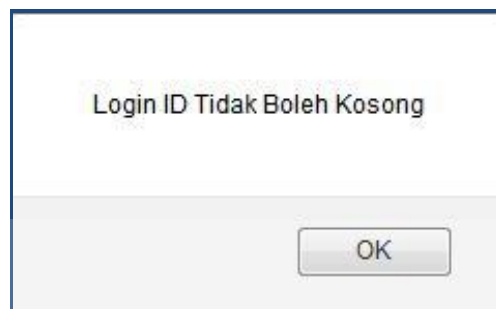
A.1.4. Uji Coba *Form Login* Pencari Kerja

Adapun proses hasil uji coba *login* pencari kerja ini bertujuan untuk mengetahui serta menentukan keberhasilan pada *form login* pencari kerja. Pada uji coba *login* pencari kerja terdapat beberapa uji coba yang dilakukan. Uji coba pertama untuk menguji kelengkapan inputan data dari *user* pencari kerja. Selanjutnya uji coba untuk menghindari salah satu data inputan salah. Uji coba yang terakhir menguji proses simpan data ketika inputan telah benar dan lengkap. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan tabel uji coba pada Tabel 4.4.

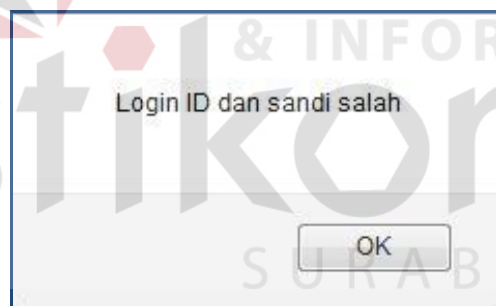
Tabel 4.4 Uji Coba *Form Login* Pencari Kerja

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
10	Menguji respon sistem ketika kolom inputan kosong	Salah satu kolom inputan data <i>login</i> pencari kerja tidak diisi	Muncul pesan kesalahan "Inputan Tidak Boleh Kosong" dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. <i>Login</i> tidak berhasil 3. Muncul pesan kesalahan "Isian Tidak Boleh Kosong"
11	Menguji respon sistem ketika salah satu data inputan salah	Salah satu kolom inputan data <i>login</i> pencari kerja diisi tidak benar	Muncul pesan kesalahan " <i>Login ID</i> dan Sandi Salah" dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. <i>Login</i> tidak berhasil 3. Muncul pesan kesalahan "<i>Login ID</i> dan Sandi Salah"
12	Menguji respon sistem ketika kolom inputan diisi dengan benar	Memasukkan data <i>login</i> pencari kerja dengan benar dan lengkap	Muncul pesan " <i>Login Berhasil</i> " dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. <i>Login</i> berhasil 3. Muncul pesan konfirmasi "<i>Login Berhasil</i>" 4. Halaman berganti

Guna menghindari kesalahan pengisian data *login* pada *form login* pencari kerja terdapat beberapa validasi yaitu validasi untuk menghindari dari data pendaftaran yang kosong dan salah satu data inputan salah akan ditampilkan pesan kesalahan seperti pada Gambar 4.39 dan Gambar 4.40 yang sesuai dengan tabel uji coba pada Tabel 4.4 nomor 10 dan nomor 11.

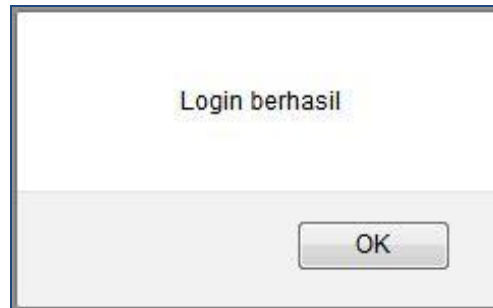


Gambar 4.39 Pesan Kesalahan Data *Login* Pencari Kerja Kosong



Gambar 4.40 Pesan Kesalahan Data *Login* Pencari Kerja Salah

Sedangkan uji coba pada Tabel 4.4 nomor 12 menghasilkan pesan konfirmasi dari data *login* penyedia kerja, setelah *user* penyedia kerja memasukkan data *login* yang benar dan lengkap akan ditampilkan pesan konfirmasi seperti pada Gambar 4.41.



Gambar 4.41 Pesan Konfirmasi *Login* Pencari Kerja Berhasil

A.1.5. Uji Coba *Form* Pencarian Lowongan Kerja

Pada uji coba *form* pencarian lowongan kerja ada beberapa uji coba yang dilakukan. Uji coba pertama untuk menguji ketika kolom pencari lowongan kerja tidak diisi. Selanjutnya uji coba untuk menguji ketika kolom pencarian lowongan kerja diisi dengan benar. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan menggunakan tabel uji coba seperti pada Tabel 4.5.

Tabel 4.5 Uji Coba *Form* Pencarian Lowongan Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
13	Menguji respon sistem ketika kolom pencarian tidak di isi	Kolom pencarian lowongan kerja tidak diisi	Muncul pesan “Hasil Pencarian Lowongan Tidak Ditemukan” dan data lowongan tidak tampil	<ol style="list-style-type: none"> 1. Sukses 2. Data pencarian kosong 3. Muncul pesan konfirmasi “Hasil Pencarian Lowongan Tidak Ditemukan”

Tabel 4.5 Uji Coba *Form* Pencarian Lowongan Kerja (Lanjutan)

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
14	Mengetahui respon sistem ketika kolom pencarian di isi	Memasukkan kolom pencarian lowongan kerja dengan benar	Muncul pesan konfirmasi “Hasil Pencarian Lowongan Yang Ditemukan Sebanyak” dan data lowongan kerja ditampilkan	<ol style="list-style-type: none"> 1. Sukses 2. Data pencarian berhasil ditampilkan 3. Muncul pesan konfirmasi “Hasil Pencarian Lowongan Yang Ditemukan Sebanyak”

Uji coba pada Tabel 4.5 nomor 13 menghasilkan pesan konfirmasi ditandai dengan tampilnya pesan seperti Gambar 4.42. Pesan konfirmasi tersebut juga akan muncul pada uji coba pada Tabel 4.5 nomor 14 seperti Gambar 4.43.



Gambar 4.42 Pesan Konfirmasi Pencarian Lowongan Kerja Tidak Ditemukan



Gambar 4.43 Pesan Konfirmasi Pencarian Lowongan Kerja Ditemukan

A.1.6. Uji Coba *Form* Data Pencari Kerja

Adapun proses hasil uji coba mengelola data pencari kerja ini bertujuan untuk mengetahui serta menentukan keberhasilan pada *form* data pencari kerja. Pada pengelolaan data pencari kerja terdapat proses yaitu proses penyimpanan untuk data pencari kerja baru dan perubahan data untuk data pencari kerja yang telah tersimpan sebelumnya, menghapus data pencari kerja dan menghindari inputan data kosong. Untuk melihat keberhasilan dalam uji coba tersebut, dilakukan pengujian dengan tabel uji coba seperti pada Tabel 4.6.

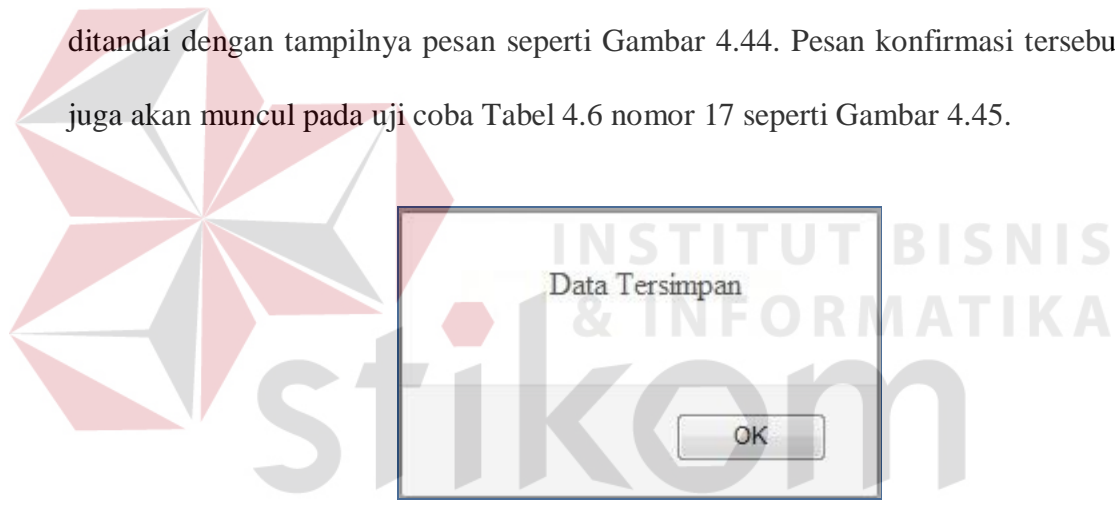
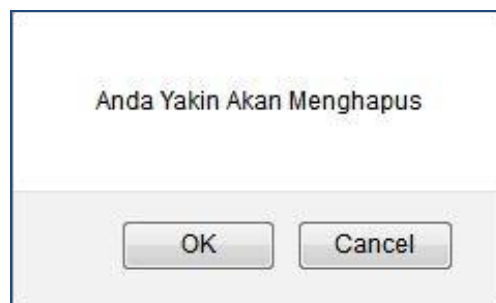
Tabel 4.6 Uji Coba *Form* Data Pencari Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
15	Menguji respon sistem ketika data pencari kerja ditambah.	Memasukkan data pencari kerja kemudian menekan tombol simpan perubahan	Muncul pesan konfirmasi “Data tersimpan” dan data telah tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil disimpan 3. Muncul pesan konfirmasi “Data Tersimpan”
16	Menguji respon sistem ketika data pencari kerja diubah	Memasukkan data pencari kerja yang baru kemudian menekan tombol simpan perubahan	Muncul pesan konfirmasi “Data tersimpan” dan data telah tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil diubah 3. Muncul pesan konfirmasi “Data Tersimpan”
17	Menguji respon sistem ketika data pencari kerja dihapus	Memilih data yang akan dihapus, kemudian menekan tombol <i>delete</i>	Muncul pesan konfirmasi “Anda Yakin Menghapus” dan data telah terhapus.	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil dihapus 3. Muncul pesan konfirmasi “Anda Yakin Menghapus”

Tabel 4.6 Uji Coba *Form* Data Pencari Kerja (Lanjutan)

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
18	Mengetahui respon sistem ketika data masukkan kosong	Salah satu masukan tidak diisi	Muncul pesan kesalahan “Inputan Harus Di isi” dan data tidak tersimpan	1. Sukses 2. Data tidak tersimpan Muncul pesan kesalahan “Inputan Harus Di isi”

Uji coba Tabel 4.6 nomor 15 dan nomor 16 menghasilkan pesan konfirmasi ditandai dengan tampilnya pesan seperti Gambar 4.44. Pesan konfirmasi tersebut juga akan muncul pada uji coba Tabel 4.6 nomor 17 seperti Gambar 4.45.

Gambar 4.44 Pesan Konfirmasi *Input* Data Pencari Kerja Tersimpan

Gambar 4.45 Pesan Konfirmasi Hapus Data Pencari Kerja

Uji coba Tabel 4.6 nomor 18 adalah memastikan validasi pengguna apakah sudah data terisi atau tidak dan jika tidak akan muncul pesan kesalahan seperti Gambar 4.46.

The image shows a web form with three input fields. Each field has a placeholder text and a red error message below it:

- No. KTP**: Placeholder "Isikan Nomer Identitas Anda", error message "No. KTP harus di isi".
- Nama Depan**: Placeholder "Isikan Nama Depan Anda", error message "Nama depan harus di isi".
- Nama Belakang**: Placeholder "Isikan Nama Belakang Anda", error message "Nama belakang harus di isi".

Gambar 4.46 Pesan Kesalahan Data Pencari Kerja Kosong

A.1.3. Uji Coba *Form* Lamar Lowongan Kerja

Proses ini bertujuan untuk mengetahui keberhasilan dari proses melamar lowongan kerja, dimana dalam proses ini dilakukan dengan cara menekan tombol lamar setelah terlebih dahulu *login* dan mengisi data pencari kerja. Untuk melihat keberhasilan dalam melakukan lamar lowongan kerja, dilakukan pengujian dengan uji coba Tabel 4.7.

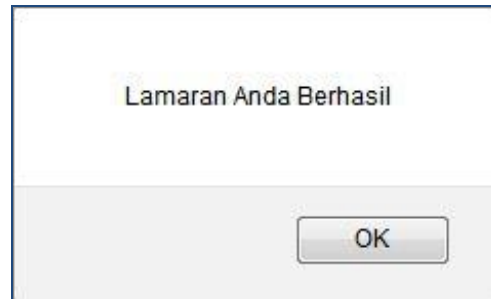
Tabel 4.7 Uji Coba *Form* Lamar Lowongan Kerja

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
19	Mengetahui respon sistem ketika melamar kerja dengan <i>login</i> data benar	Memasukkan lamar kerja dengan terlebih dahulu <i>login</i> kemudian menekan tombol lamar	Muncul pesan konfirmasi "Lamaran Anda Berhasil" dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil disimpan 3. Muncul pesan "Lamaran Anda Berhasil"

Tabel 4.7 Uji Coba *Form* Lamar Lowongan Kerja (Lanjutan)

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
20	Mengetahui respon sistem ketika melamar kerja tetapi belum <i>login</i>	Memasukkan lamar kerja dengan tidak terlebih dahulu <i>login</i> kemudian menekan tombol lamar	Muncul pesan kesalahan “Anda Harus <i>Login</i> Terlebih Dahulu” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak disimpan 3. Muncul pesan kesalahan “Anda Harus <i>Login</i> Terlebih Dahulu”
21	Mengetahui respon sistem ketika melamar kerja pada lowongan yang sama	Memasukkan lamaran kerja pada data lowongan yang sama	Muncul pesan kesalahan “Anda Sudah Pernah Melamar” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak disimpan 3. Muncul pesan kesalahan “Anda Sudah Pernah Melamar”
22	Mengetahui respon sistem ketika lamaran tidak sesuai kualifikasi	Memasukkan lamaran pada lowongan yang tidak sesuai kualifikasi	Muncul pesan kesalahan “Anda Belum Sesuai Dengan Kebutuhan Pada Posisi Ini” dan data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak disimpan 3. Muncul pesan kesalahan “Anda Belum Sesuai Dengan Kebutuhan Pada Posisi Ini”
23	Mengetahui respon sistem ketika membatalkan lamaran kerja	Memilih data lowongan kemudian menekan tombol Batal	Data tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak tersimpan 3. <i>Form</i> kembali ke halaman awal

Uji coba pada Tabel 4.7 nomor 19 menghasilkan pesan konfirmasi dari data lamar yang ditandai dengan tampilnya pesan seperti Gambar 4.47.

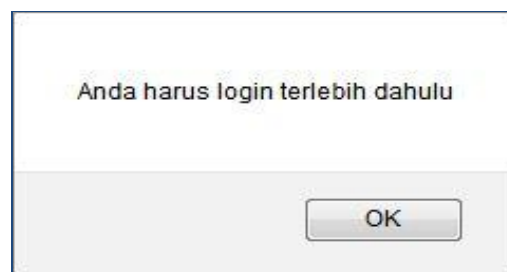


Gambar 4.47 Pesan Konfirmasi Lamaran Lowongan Kerja Berhasil

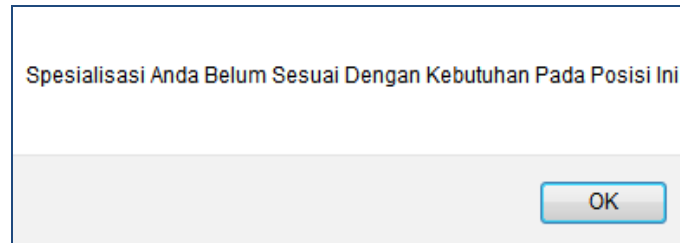
Uji coba pada Tabel 4.7 nomor 21 adalah memastikan validasi pengguna apakah sudah pernah melamar pada lowongan kerja yang sama atau tidak, jika sudah akan muncul pesan kesalahan seperti Gambar 4.48. Pesan validasi tersebut juga akan muncul pada uji coba Tabel 4.7 nomor 20 dan 22 seperti Gambar 4.49 dan Gambar 4.50.



Gambar 4.48 Pesan Kesalahan Sudah Pernah Melamar



Gambar 4.49 Pesan Kesalahan Belum *Login*



Gambar 4.50 Pesan Kesalahan Lamaran Tidak Sesuai

A.1.4. Uji Coba *Form* Membuat Lowongan Kerja

Proses ini bertujuan untuk mengetahui keberhasilan dari proses buat lowongan kerja, dimana dalam proses ini dilakukan dengan cara menginputkan beberapa inputan oleh *user*. Untuk melihat keberhasilan dalam melakukan pembuatan lowongan kerja baru, dilakukan uji coba pada Tabel 4.8.

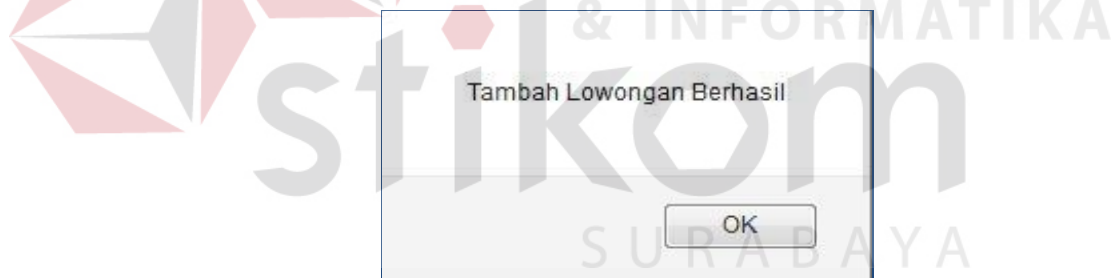
Tabel 4.8 Uji Coba *Form* Membuat Lowongan Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
24	Mengetahui respon ketika dilakukan tambah data lowongan kerja baru	Memasukkan lowongan baru kemudian menekan tombol simpan	Muncul pesan konfirmasi "Tambah Lowongan Berhasil" dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil disimpan 3. Muncul pesan konfirmasi "Tambah Lowongan Berhasil"
25	Mengetahui respon ketika mengubah data lowongan kerja	Memasukkan data baru kemudian menekan tombol simpan	Muncul pesan konfirmasi "Ubah Lowongan Berhasil" dan data tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil diubah 3. Muncul pesan konfirmasi "Ubah Lowongan Berhasil"

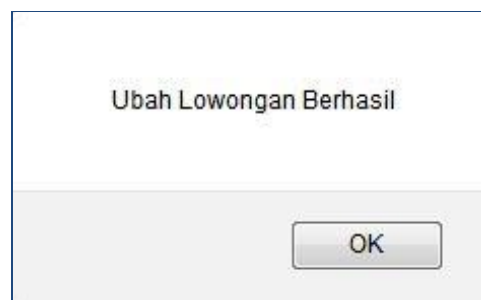
Tabel 4.8 Uji Coba *Form* Membuat Lowongan Kerja (Lanjutan)

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
26	Mengetahui respon sistem ketika data lowongan kerja dihapus	Memilih data yang akan dihapus, kemudian menekan tombol <i>delete</i>	Muncul pesan “Anda Yakin Menghapus” dan data telah terhapus.	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil dihapus 3. Muncul pesan “Anda Yakin Menghapus”

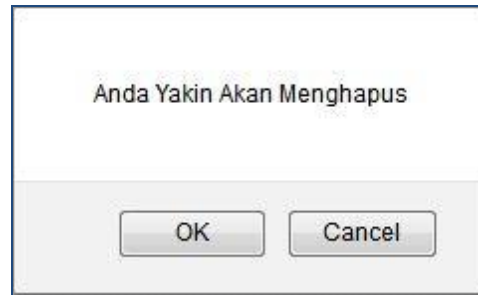
Uji coba pada Tabel 4.8 nomor 24 menghasilkan pesan konfirmasi dari data lowongan kerja yang ditandai dengan tampilnya pesan seperti Gambar 4.51. Pesan konfirmasi tersebut juga akan muncul pada uji coba pada Tabel 4.8 nomor 25 dan nomor 26 seperti pada Gambar 4.52 dan Gambar 4.53.



Gambar 4.51 Pesan Konfirmasi Membuat Lowongan Kerja Berhasil



Gambar 4.52 Pesan Konfirmasi Ubah Lowongan Kerja Berhasil



Gambar 4.53 Pesan Konfirmasi Hapus Lowongan Kerja

A.1.5. Uji Coba *Form Bobot Kriteria*

Proses ini bertujuan untuk mengetahui keberhasilan dari proses input bobot kriteria, dimana dalam proses ini penentuan bobot kriteria ini dilakukan dengan cara menginputkan bobot pada setiap kriteria seleksi oleh *user* penyedia kerja. Untuk melihat keberhasilan dalam melakukan input bobot kriteria, dilakukan pengujian dengan uji coba pada Tabel 4.9.

Tabel 4.9 Uji Coba *Form Bobot Kriteria*

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
27	Menguji respon sistem ketika data bobot kriteria seleksi ditambah	Memasukkan bobot kriteria seleksi kemudian menekan tombol tambah nilai dan simpan	Data nilai bobot berhasil ditampilkan pada <i>datagrid</i>	<ol style="list-style-type: none"> 1. Sukses 2. Data bobot kriteria seleksi berhasil ditambah dan ditampilkan pada <i>datagrid</i>
28	Menguji respon sistem ketika inputan data bobot kriteria seleksi bernilai sama atau dalam nilai <i>range</i>	Memasukkan data bobot kriteria seleksi yang telah ada atau dalam nilai <i>range</i>	Data nilai bobot sama tidak berhasil ditampilkan pada <i>datagrid</i>	<ol style="list-style-type: none"> 1. Sukses 2. Isian data nilai bobot kriteria tidak berhasil 3. <i>Form</i> kembali ke halaman awal

Tabel 4.9 Uji Coba *Form* Bobot Kriteria (Lanjutan)

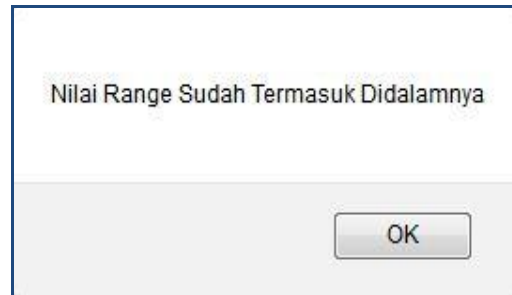
Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
29	Menguji respon sistem ketika data bobot kriteria seleksi dihapus	Memilih data bobot kriteria seleksi yang ada, kemudian menekan tombol <i>delete</i>	Muncul pesan konfirmasi “Anda Yakin Menghapus” dan data bobot terhapus dari <i>datagrid</i> .	<ol style="list-style-type: none"> 1. Sukses 2. Data berhasil dihapus 3. Muncul pesan konfirmasi “Anda Yakin Menghapus”
30	Membatalkan penyimpanan dan perubahan data	Memasukkan data nilai bobot kriteria kemudian menekan tombol Batal	Semua kolom kosong dan tidak tersimpan	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak tersimpan 3. <i>Form</i> kembali ke halaman awal

Uji coba pada Tabel 4.9 nomor 27 berhasil dengan ditandai tampilnya data bobot kriteria seleksi pada *datagrid* seperti Gambar 4.54.

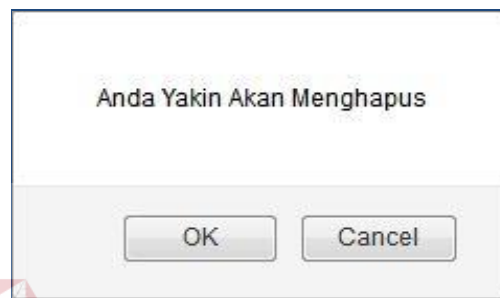
No	Jenis Kriteria	Nilai Min	Nilai Max	Bobot	Aksi
1	Usia	45	64	0.00	<input type="button" value="DEL"/>
2	Usia	35	44	0.33	<input type="button" value="DEL"/>
3	Usia	27	34	0.67	<input type="button" value="DEL"/>
4	Usia	20	26	1.00	<input type="button" value="DEL"/>

Gambar 4.54 Tampilan Nilai Bobot Kriteria Seleksi

Uji coba pada Tabel 4.9 nomor 28 dan 29 berhasil ditandai dengan tampilnya pesan konfirmasi seperti Gambar 4.55 dan 4.56



Gambar 4.55 Pesan Konfirmasi Data Bobot Bernilai Sama



Gambar 4.56 Pesan Konfirmasi Hapus Data Bobot Kriteria Seleksi

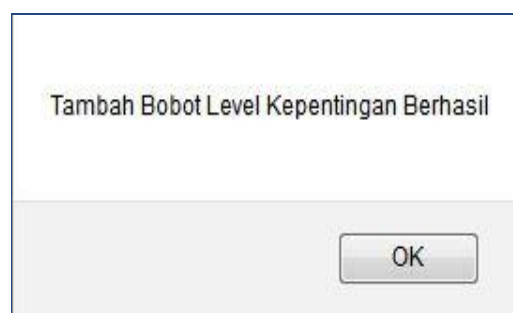
A.1.6. Uji Coba *Form* Bobot Kepentingan

Proses ini bertujuan untuk mengetahui keberhasilan dari proses membuat bobot kepentingan (W), dimana dalam proses ini dilakukan dengan cara menginputkan beberapa inputan oleh *user* penyedia kerja. Untuk melihat keberhasilan dalam melakukan input bobot level kepentingan, dilakukan pengujian dengan uji coba seperti pada Tabel 4.10.

Tabel 4.10 Uji Coba *Form* Bobot Kepentingan

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
31	Menguji respon sistem ketika bobot kepentingan ditambah	Memasukkan nilai bobot kepentingan kemudian menekan tombol simpan.	Muncul pesan “Tambah Bobot Level Kepentingan Berhasil” dan inputan diubah ke dalam bilangan <i>fuzzy</i> serta data tersimpan.	<ol style="list-style-type: none"> 1. Sukses 2. Bobot level kepentingan berhasil di ubah ke dalam bilangan <i>fuzzy</i> 3. Muncul pesan “Tambah Bobot Level Kepentingan Berhasil”
32	Membatalkan penyimpanan masukkan data.	Memasukkan bobot kepentingan kemudian menekan tombol batal.	Semua kolom kosong dan data tidak tersimpan.	<ol style="list-style-type: none"> 1. Sukses 2. Data tidak tersimpan 3. <i>Form</i> kembali ke halaman awal
33	Menghindari kekosongan data masukkan.	Salah satu atau semua kolom masukan data tidak diisi.	Semua kolom yang kosong bernilai 0.	<ol style="list-style-type: none"> 1. Sukses 2. Data nilai bobot level kepentingan bernilai 0

Uji coba pada Tabel 4.9 nomor 31 berhasil ditandai dengan tampilnya pesan konfirmasi seperti Gambar 4.57.



Gambar 4.57 Pesan Konfirmasi Tambah Bobot Kepentingan Berhasil

A.1.7. Uji Coba *Form Filtering* Pencari Kerja

Proses ini bertujuan untuk mengetahui keberhasilan dari proses *filtering* pencari kerja, dimana dalam proses ini dilakukan dengan cara menginputkan beberapa kategori *filtering* yang ada oleh *user* penyedia kerja. Untuk melihat keberhasilan dalam melakukan *filtering* pencari kerja, dilakukan pengujian dengan uji coba seperti pada Tabel 4.11.

Tabel 4.11 Uji Coba *Form Filtering* Pencari Kerja

Test Case ID	Tujuan	Input	<i>Output</i> Diharapkan	<i>Output</i> Sistem
34	Menguji respon sistem ketika kolom <i>filtering</i> kosong	Kolom <i>filtering</i> pencari kerja tidak diisi	Muncul pesan konfirmasi “Data Pencari Kerja tidak tampil” dan data Pencari Kerja tidak tampil	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>filtering</i> kosong 3. Muncul pesan konfirmasi “Data Pencari Kerja Tidak Ditemukan”
35	Menguji respon sistem ketika data pencarian diinputkan dengan benar	Salah satu kolom kategori <i>filtering</i> pencari kerja diisi	Data <i>filtering</i> Pencari Kerja berhasil ditampilkan pada <i>datagrid</i>	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>filtering</i> Pencari Kerja berhasil ditampilkan pada <i>datagrid</i>
36	Menguji respon sistem ketika data pencarian dibatalkan	Memilih kolom kategori <i>filtering</i> pencari kerja	Pencarian tidak diproses	<ol style="list-style-type: none"> 1. Sukses 2. Data pencarian dibatalkan 3. Data <i>filtering</i> tidak diproses

Uji coba pada Tabel 4.11 nomor 34 berhasil ditandai dengan tampilnya pesan konfirmasi seperti Gambar 4.58.

Nama	Kelamin	Usia	Kota	Propinsi	IPK	Akre	Pengalaman	Jabatan	Aksi
Data Pelamar Tidak Ditemukan									

Gambar 4.58 Pesan Konfirmasi *Filtering* Pencari Kerja Tidak Ditemukan

Uji coba pada Tabel 4.11 nomor 35 berhasil ditandai dengan tampilnya data *filtering* pencari kerja pada *datagrid* seperti Gambar 4.59.

Usia	IPK	Propinsi	Pengalaman Kerja							
<input type="text"/> Min <input type="text"/> Max	<input type="text"/> Min <input type="text"/> Max	--	-- Tahun							
Jenis Kelamin	Akreditasi	Kota	Jabatan							
Laki-laki	--	--	--							
<input type="button" value="Cari"/> <input type="button" value="Batal"/>										
Halaman : 1 2										
Total Jumlah Pelamar : 10										
Nama	Lowongan	Kelamin	Usia	Kota	Propinsi	IPK	Akre	Pengalaman	Jabatan	Aksi
Fanca Variska	Civil And Structural Engineer	Laki-laki	26 Tahun	Surabaya	Jawa timur	3.75	B	1 Tahun	General Affair / Bagian Umum	<input type="button" value="Compare"/>
Supriyadi Asmanto	Marketing	Laki-laki	28 Tahun	Surabaya	Jawa timur	3.80	A	5 Tahun	Staf Administrasi	<input type="button" value="Compare"/>
Bekti Indro	Civil And Structural Engineer	Laki-laki	28 Tahun	Surabaya	Jawa timur	2.80	A	4 Tahun	General Affair / Bagian Umum	<input type="button" value="Compare"/>
Agus Hafiludin	Marketing	Laki-laki	33 Tahun	Sidoarjo	Jawa timur	3.20	A	3 Tahun	Staf Administrasi	<input type="button" value="Compare"/>

Gambar 4.59 Tampilan *Filtering* Pencari Kerja Berhasil

A.1.8. Uji Coba *Form Compare* Pencari Kerja

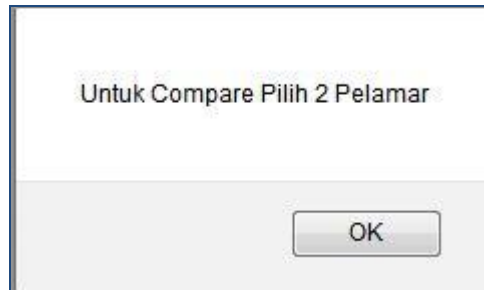
Proses ini bertujuan untuk mengetahui keberhasilan dari proses *compare* pencari kerja, dimana dalam proses ini dilakukan dengan cara menginputkan data pencari kerja yang telah melamar. Proses *compare* pencari kerja terbagi menjadi dua bagian yaitu *compare* khusus atau membandingkan data pencari kerja yang melamar dalam lowongan yang sama dan *compare* umum atau membandingkan data pencari kerja yang melamar secara global. Untuk melihat keberhasilan dalam

melakukan *compare* pencari kerja, dilakukan pengujian dengan uji coba seperti pada Tabel 4.12.

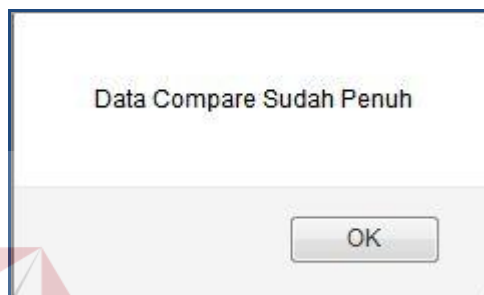
Tabel 4.12 Uji Coba *Form Compare* Pencari Kerja

Test Case ID	Tujuan	Input	Output Diharapkan	Output Sistem
37	Menguji respon sistem ketika data <i>compare</i> kosong atau terdapat satu data <i>compare</i> pencari kerja	Memilih satu data pencari kerja kemudian menekan button <i>Compare</i> lalu menekan tombol <i>Compare All</i>	Muncul pesan kesalahan “Untuk <i>Compare</i> pilih 2 Pencari Kerja” dan data <i>compare</i> pencari kerja tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>compare</i> tidak berhasil 3. Muncul pesan kesalahan “Untuk <i>Compare</i> pilih 2 Pencari Kerja”
38	Menguji respon sistem ketika memasukkan lebih dari 2 data <i>compare</i> pencari kerja	Menekan button <i>compare</i> lebih dari 2 data pencari kerja	Muncul pesan kesalahan “Data <i>Compare</i> sudah penuh” dan data <i>compare</i> pencari kerja tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>compare</i> tidak berhasil 3. Muncul pesan kesalahan “Data <i>Compare</i> sudah penuh”
39	Menguji respon sistem ketika memasukkan data <i>compare</i> pencari kerja sama	Memilih data pencari kerja yang sama kemudian menekan button <i>Compare</i>	Muncul pesan kesalahan “Harap Pilih Data Lain” dan data <i>compare</i> pencari kerja tidak berhasil	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>compare</i> tidak berhasil 3. Muncul pesan kesalahan “Harap Pilih Data Lain”
40	Menguji respon sistem ketika inputan data <i>compare</i> sudah benar dan lengkap	Memilih 2 data pencari kerja untuk dilakukan perbandingan kemudian menekan button <i>Compare</i>	Data <i>compare</i> berhasil ditampilkan	<ol style="list-style-type: none"> 1. Sukses 2. Data <i>compare</i> berhasil ditampilkan

Uji coba pada Tabel 4.12 nomor 37 dan 38 berhasil ditandai dengan tampilnya pesan kesalahan seperti Gambar 4.60 dan Gambar 4.61.



Gambar 4.60 Pesan Kesalahan Data *Compare* Pencari Kerja



Gambar 4.61 Pesan Kesalahan Data *Compare* Pencari Kerja Sudah Terisi

Uji coba pada Tabel 4.12 nomor 39 berhasil ditandai dengan tampilnya pesan kesalahan seperti Gambar 4.62.



Gambar 4.62 Pesan Kesalahan Data *Compare* Pencari Kerja Sama

Sedangkan pada uji coba pada Tabel 4.12 nomor 40 berhasil ditandai dengan tampilnya data *compare* pencari kerja pada *datagrid* seperti Gambar 4.63.



Head To Head		
Pelamar 1	VS	Pelamar 2
Catur Wicaksono	Nama	Fanca Variska
Laki-laki	Jenis Kelamin	Laki-laki
1988-07-23	Kelahiran	1988-06-15
View Details		View Details

Gambar 4.63 Tampilan *Compare* Pencari Kerja Berhasil

B. Uji Coba Perhitungan Sistem

Analisa hasil uji coba dilakukan untuk menguji kinerja sistem untuk mencari pencari kerja terbaik berdasarkan perhitungan bobot dan perbandingan.

Tujuan analisa hasil uji coba ini untuk mencari keakuratan kinerja sistem dalam proses perhitungan menggunakan metode *Simple Additive Weighting*.

A. Uji Kasus

Berdasarkan tujuan pengujian untuk mencari pencari kerja terbaik, maka terdapat kasus uji untuk mengetahui hasil perhitungan tertulis dengan perhitungan sistem. Berikut penjelasan kasus uji dan data uji mencari pencari kerja terbaik.

Uji kasus ini adalah penentuan alternatif pencari kerja terbaik untuk lowongan kerja *Dekstop Programmer* pada PT. Mekanika. Sebelum melakukan perhitungan menggunakan metode *Simple Additive Weighting* akan ditetapkan terlebih dahulu oleh sistem adalah sebagai berikut :

1. Kriteria-kriteria seleksi (C) antara lain kriteria usia, pendidikan terakhir, akreditasi, IPK/NEM, pengalaman kerja, tinggi badan, berat badan, status dan kondisi mata.
2. Jenis atribut (atribut keuntungan/*benefit* ataupun atribut biaya/*cost*) seperti yang ditunjukkan pada Tabel 4.12.

3. Bobot kepentingan setiap alternatif pada setiap kriteria yang terlebih dahulu dikonversikan ke dalam bilangan *fuzzy* dengan menggunakan perhitungan variabel ke- n / $n-1$. Data bobot kepentingan tampak pada Tabel 4.13.

Tabel 4.13 Jenis Atribut

Kode	Nama Kriteria	Atribut
C1	Usia	<i>Cost</i>
C2	Pendidikan	<i>Benefit</i>
C3	Akreditasi	<i>Benefit</i>
C4	IPK	<i>Benefit</i>
C5	Pengalaman Kerja	<i>Benefit</i>
C6	Tinggi Badan	<i>Benefit</i>
C7	Berat Badan	<i>Cost</i>
C8	Status	<i>Cost</i>
C9	Kondisi Mata	<i>Cost</i>

Tabel 4.14 Bobot Kepentingan

Variabel	Bobot (Nilai)
Sangat Tidak Penting	Variabel ke-0 / (5-1) = 0/4 = 0
Tidak Penting	Variabel ke-1 / (5-1) = 1/4 = 0.25
Cukup Penting	Variabel ke-2 / (5-1) = 2/4 = 0.5
Penting	Variabel ke-3 / (5-1) = 3/4 = 0.75
Sangat Penting	Variabel ke-4 / (5-1) = 4/4 = 1

A1. Input Keterangan Bobot Kriteria

Berdasarkan kriteria, jenis atribut dan bobot kepentingan setiap alternatif pada setiap kriteria yang telah ditentukan. Maka selanjutnya perlu di inputkan keterangan bobot kriteria pada setiap kriteria penyeleksian yang ada.

1. Kriteria Usia

Usia pencari kerja dinilai berdasarkan dari batas usia pencari kerja tersebut. Usia memiliki jenis atribut *cost* karena semakin minimal atau muda usia pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin maksimal atau tua usia pencari kerja semakin berkurang nilainya.

Tabel 4.15 Data Nilai Bobot Kriteria Usia

Variabel Usia	Bobot (Nilai)
Usia 45-64	Variabel ke $- 0 / (4-1) = 0/3 = 0$
Usia 35-44	Variabel ke $- 1 / (4-1) = 1/3 = 0,33$
Usia 25-34	Variabel ke $- 2 / (4-1) = 2/3 = 0,67$
Usia 20-24	Variabel ke $- 3 / (4-1) = 3/3 = 1$



No	Jenis Kriteria	Nilai Min	Nilai Max	Bobot	Aksi
1	Usia	45	64	0.00	DEL
2	Usia	35	44	0.33	DEL
3	Usia	25	34	0.67	DEL
4	Usia	20	24	1.00	DEL

Gambar 4.64 Hasil *Input* Nilai Bobot Kriteria Usia

2. Kriteria Pendidikan

Pendidikan pencari kerja dinilai berdasarkan dari pendidikan terakhir pencari kerja tersebut. Pendidikan memiliki jenis atribut *benefit* karena semakin maksimal atau tinggi pendidikan pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin minimal atau rendah pendidikan pencari kerja semakin berkurang nilainya.

Tabel 4.16 Data Nilai Bobot Kriteria Pendidikan

Variabel Pendidikan	Bobot (Nilai)
SMA	Variabel ke $0 / (5-1) = 0/4 = 0$
D3	Variabel ke $1 / (5-1) = 1/4 = 0,25$
S1	Variabel ke $2 / (5-1) = 2/4 = 0,5$
S2	Variabel ke $3 / (5-1) = 3/4 = 0,75$
S3	Variabel ke $4 / (5-1) = 4/4 = 1$

No	Jenis Kriteria	Keterangan (Nilai)	Bobot	Aksi
1	Pendidikan	SMA	0.00	<input type="button" value="OK"/>
2	Pendidikan	D3	0.25	<input type="button" value="OK"/>
3	Pendidikan	S1	0.50	<input type="button" value="OK"/>
4	Pendidikan	S2	0.75	<input type="button" value="OK"/>
5	Pendidikan	S3	1.00	<input type="button" value="OK"/>





Gambar 4.65 Hasil *Input* Nilai Bobot Kriteria Pendidikan

3. Kriteria Akreditasi

Akreditasi dinilai berdasarkan dari status akreditasi instansi yang menjadi tempat pencari kerja tersebut menimba ilmu. Akreditasi memiliki jenis atribut *benefit* karena semakin maksimal atau tinggi akreditasi instansi semakin bagus nilainya, dan begitu pula sebaliknya semakin minimal atau rendah akreditasi instansi pencari kerja semakin berkurang nilainya.

Tabel 4.17 Data Nilai Bobot Kriteria Akreditasi

Variabel Akreditasi	Bobot (Nilai)
Tidak Terakreditasi	Variabel ke $0 / (4-1) = 0/3 = 0$
C	Variabel ke $1 / (4-1) = 1/3 = 0,33$
B	Variabel ke $2 / (4-1) = 2/3 = 0,67$
A	Variabel ke $3 / (4-1) = 3/3 = 1$

No	Jenis Kriteria	Keterangan (Nilai)	Bobot	Aksi
1	Akreditasi	Tidak Terakreditasi	0,00	
2	Akreditasi	C	0,33	
3	Akreditasi	B	0,67	
4	Akreditasi	A	1,00	

Gambar 4.66 Hasil *Input* Nilai Bobot Kriteria Akreditasi

4. Kriteria IPK

IPK pencari kerja dinilai berdasarkan dari nilai prestasi pencari kerja selama menempuh studi. IPK memiliki jenis atribut *benefit* karena semakin maksimal atau tinggi nilai prestasi pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin minimal atau rendah nilai prestasi pencari kerja semakin berkurang nilainya.

Tabel 4.18 Data Nilai Bobot Kriteria IPK

Variabel IPK	Bobot (Nilai)
IPK < 2,75	Variabel ke - 0 / (6-1) = 0/5 = 0
IPK 2,75 – 3,00	Variabel ke - 1 / (6-1) = 1/5 = 0,2
IPK 3,01 – 3,25	Variabel ke - 2 / (6-1) = 2/5 = 0,4
IPK 3,26 – 3,50	Variabel ke - 3 / (6-1) = 3/5 = 0,6
IPK 3,51 – 3,75	Variabel ke - 4 / (6-1) = 4/5 = 0,8
IPK 3,76 – 4,00	Variabel ke - 5 / (6-1) = 5/5 = 1

No	Jenis Kriteria	Nilai Min	Nilai Max	Bobot	Aksi
1	IPK	2.5	2.74	0.00	DEL
2	IPK	2.75	3	0.20	DEL
3	IPK	3.01	3.25	0.40	DEL
4	IPK	3.26	3.5	0.60	DEL
5	IPK	3.51	3.75	0.80	DEL
6	IPK	3.76	4	1.00	DEL

Gambar 4.67 Hasil *Input* Nilai Bobot Kriteria IPK

5. Kriteria Pengalaman Kerja

Pengalaman kerja pencari kerja dinilai berdasarkan dari riwayat pengalaman kerja dari pencari kerja. Pengalaman kerja memiliki jenis atribut *benefit* karena semakin maksimal atau tinggi pengalaman kerja pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin minimal atau rendah pengalaman kerja pencari kerja semakin berkurang nilainya.

Tabel 4.19 Data Nilai Bobot Kriteria Pengalaman Kerja

Variabel Pengalaman	Bobot (Nilai)
0 tahun	Variabel ke $- 0 / (8-1) = 0/7 = 0$
1 tahun	Variabel ke $- 1 / (8-1) = 1/7 = 0,14$
2 tahun	Variabel ke $- 2 / (8-1) = 2/7 = 0,29$
3 tahun	Variabel ke $- 3 / (8-1) = 3/7 = 0,43$
4 tahun	Variabel ke $- 4 / (8-1) = 4/7 = 0,57$
5 tahun	Variabel ke $- 5 / (8-1) = 5/7 = 0,71$
6 tahun	Variabel ke $- 6 / (8-1) = 6/7 = 0,86$
7 tahun	Variabel ke $- 7 / (8-1) = 7/7 = 1$

No	Jenis Kriteria	Keterangan (Nilai)	Bobot	Aksi
1	Pengalaman	0	0.00	DEL
2	Pengalaman	1	0.14	DEL
3	Pengalaman	2	0.29	DEL
4	Pengalaman	3	0.43	DEL
5	Pengalaman	4	0.57	DEL
6	Pengalaman	5	0.71	DEL
7	Pengalaman	6	0.86	DEL
8	Pengalaman	7	1.00	DEL

Gambar 4.68 Hasil *Input* Nilai Bobot Kriteria Pengalaman Kerja

6. Kriteria Tinggi Badan

Tinggi badan pencari kerja dinilai berdasarkan dari tinggi badan ideal dan proporsional dari pencari kerja. Tinggi badan memiliki jenis atribut *benefit* karena semakin maksimal tinggi badan pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin minimal tinggi badan pencari kerja semakin berkurang nilainya.

Tabel 4.20 Data Nilai Bobot Kriteria Tinggi Badan

Variabel Tinggi Badan	Bobot (Nilai)
<147	Variabel ke – 0 / (6-1) = 0/ 5 = 0
148-153	Variabel ke – 1 / (6-1) = 1/5 = 0,2
154-159	Variabel ke – 2 / (6-1) = 2/5 = 0,4
160-165	Variabel ke – 3 / (6-1) = 3/5 = 0,6
166-171	Variabel ke – 4 / (6-1) = 4/5 = 0,8
172-180	Variabel ke – 5 / (6-1) = 5/5 = 1

No	Jenis Kriteria	Nilai Min	Nilai Max	Bobot	Aksi
1	TinggiBadan	140	147	0.00	DEL
2	TinggiBadan	148	153	0.20	DEL
3	TinggiBadan	154	159	0.40	DEL
4	TinggiBadan	160	165	0.60	DEL
5	TinggiBadan	166	171	0.80	DEL
6	TinggiBadan	172	180	1.00	DEL

Gambar 4.69 Hasil *Input* Nilai Bobot Kriteria Tinggi Badan

7. Kriteria Berat Badan

Berat badan pencari kerja dinilai berdasarkan dari berat badan ideal dan proporsional dari pencari kerja. Berat badan memiliki jenis atribut *cost* karena semakin maksimal berat badan pencari kerja semakin kurang nilainya, dan begitu pula sebaliknya semakin minimal berat badan pencari kerja semakin bagus nilainya.

Tabel 4.21 Data Nilai Bobot Kriteria Berat Badan

Variabel Tinggi Badan	Bobot (Nilai)
85-150	Variabel ke $- 0 / (6-1) = 0/ 5 = 0$
76-84	Variabel ke $- 1 / (6-1) = 1/5 = 0,2$
65-75	Variabel ke $- 2 / (6-1) = 2/5 = 0,4$
60-64	Variabel ke $- 3 / (6-1) = 3/5 = 0,6$
56-59	Variabel ke $- 4 / (6-1) = 4/5 = 0,8$
50-55	Variabel ke $- 5 / (6-1) = 5/5 = 1$

No	Jenis Kriteria	Nilai Min	Nilai Max	Bobot	Aksi
1	BeratBadan	85	150	0.00	DEL
2	BeratBadan	76	84	0.20	DEL
3	BeratBadan	65	75	0.40	DEL
4	BeratBadan	60	64	0.60	DEL
5	BeratBadan	56	59	0.80	DEL
6	BeratBadan	50	55	1.00	DEL

Gambar 4.70 Hasil *Input* Nilai Bobot Kriteria Berat Badan

8. Kriteria Status Pernikahan

Status pencari kerja dinilai berdasarkan dari status pernikahan dari pencari kerja. Status pernikahan memiliki jenis atribut *cost* karena semakin status belum menikah pencari kerja semakin bagus nilainya, dan begitu pula sebaliknya semakin status menikah pencari kerja semakin berkurang nilainya.

Tabel 4.22 Data Nilai Bobot Kriteria Status Pernikahan

Variabel Status	Bobot (Nilai)
Menikah	Variabel ke $- 0 / (2-1) = 0/1 = 0$
Belum Menikah	Variabel ke $- 1 / (2-1) = 1/1 = 1$

No	Jenis Kriteria	Keterangan (Nilai)	Bobot	Aksi
1	StatusNikah	Menikah	0.00	DEL
2	StatusNikah	Belum Menikah	1.00	DEL

Gambar 4.71 Hasil *Input* Nilai Bobot Kriteria Status Pernikahan

9. Kriteria Kondisi Mata

Kondisi mata pencari kerja dinilai berdasarkan dari kondisi mata dari pencari kerja. Kondisi mata memiliki jenis atribut *cost* karena semakin pencari kerja tidak menggunakan kacamata semakin bagus nilainya, dan begitu pula sebaliknya semakin pencari kerja berkacamata semakin berkurang nilainya.

Tabel 4.23 Data Nilai Bobot Kriteria Kondisi Mata

Variabel Kondisi Mata	Bobot (Nilai)
Berkacamata	Variabel ke $- 0 / (2-1) = 0$
Tidak Berkacamata	Variabel ke $- 1 / (2-1) = 1/1 = 1$



No	Jenis Kriteria	Keterangan (Nilai)	Bobot	Aksi
1	KondisiMata	Ya	0.00	DEL
2	KondisiMata	Tidak	1.00	DEL

Gambar 4.72 Hasil *Input* Nilai Bobot Kriteria Kondisi Mata

Berikut adalah Tabel inputan data diri pencari kerja yang telah melamar pada lowongan kerja Dekstop Programmer dan merupakan data awal dari perhitungan metode *simple additive weighting* (SAW).

Tabel 4.24 Data Inputan Pencari Kerja

Alternatif	Kriteria								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
A1	26	D3	B	2.75	4 Tahun	167	55	Belum Menikah	Tidak Berkacamata
A2	25	S1	B	3.25	1 Tahun	168	55	Belum Menikah	Tidak Berkacamata

Tabel 4.24 Data Inputan Pencari Kerja (Lanjutan)

Alternatif	Kriteria								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
A3	24	S1	A	4	6 Tahun	165	67	Belum Menikah	Tidak Berkacamata
A4	28	S1	C	2.45	0 Tahun	172	50	Menikah	Berkacamata

Keterangan :

A. Kriteria

1. C1 = Usia
2. C2 = Pendidikan Terakhir
3. C3 = Akreditasi
4. C4 = IPK / NEM
5. C5 = Pengalaman Kerja
6. C6 = Tinggi Badan
7. C7 = Berat Badan
8. C8 = Status Menikah
9. C9 = Kondisi Mata

B. Alternatif Pencari Kerja

1. A1 = Arya
2. A2 = Jodi
3. A3 = Edo
4. A4 = Dono

Berikut adalah Tabel data pencari kerja yang kemudian dibuatkan sebuah matriks keputusan ternormalisasi (X) dengan sebelumnya bahwa bobot setiap kriteria tersebut telah dikonversikan dengan bilangan *fuzzy*.

Tabel 4.25 Data Matriks Keputusan (X)

Alternatif	Kriteria								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
A1	0.67	0.25	0.67	0.2	0.57	0.8	1	1	1
A2	0.67	0.5	0.67	0.4	0.14	0.8	1	1	1
A3	1	0.5	1	1	0.86	0.6	0.4	1	1
A4	0.67	0.5	0.33	0	0	1	1	0	0

Informasi data matriks keputusan (X) pada aplikasi yang merupakan data awal pada perhitungan metode *simple additive weighting* dapat dilihat pada Gambar 4.73.

Email Pelamar	Nama Pelamar	Nilai Matriks Keputusan Kriteria Seleksi (X)								
		Usia	Pendidikan	Akreditasi	IPK	Pengalaman	Tinggi	Berat	Status	Kacamata
arya@gmail.com	Arya Prabowo	0.67	0.25	0.67	0.20	0.57	0.80	1.00	1.00	1.00
Dono@gmail.com	Dono Iswanto	0.67	0.50	0.33	0.00	0.00	1.00	1.00	0.00	0.00
edo@gmail.com	Edo Tambun	1.00	0.50	1.00	1.00	0.86	0.60	0.40	1.00	1.00
jodi@gmail.com	Jodi Setiawan	0.67	0.50	0.67	0.40	0.14	0.80	1.00	1.00	1.00

Gambar 4.73 Uji Coba Menampilkan Data Matriks Keputusan (X)

Adapun langkah berikutnya dalam perhitungan metode *simple additive weighting* adalah menentukan bobot variabel kepentingan (W) dari masing-

masing kriteria seleksi. Untuk contoh kasus ini, berikut adalah nilai bobot yang telah ditentukan:

Tabel 4.26 Data Bobot Kepentingan

Kriteria	Bobot (nilai)	Keterangan
C1	5	Sangat Penting
C2	5	Sangat Penting
C3	5	Sangat Penting
C4	5	Sangat Penting
C5	5	Sangat Penting
C6	5	Sangat Penting
C7	5	Sangat Penting
C8	5	Sangat Penting
C9	5	Sangat Penting

Pada aplikasi, data bobot kepentingan (W) dapat ditunjukkan pada Gambar 4.74 sebagai berikut :

Usia	5
Pendidikan	5
Akreditasi	5
IPK	5
Pengalaman	5
Tinggi Badan	5
Berat Badan	5
Status Menikah	5
Berkacamata	5

Gambar 4.74 Uji Coba Menampilkan Data Bobot Kepentingan

Langkah kedua yang dilakukan adalah menormalisasi matriks keputusan (X) menjadi matriks normalisasi (R) dengan menggunakan persamaan sebagai berikut

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{x_{ij}}{\min x_{ij}} & \text{jika } j \text{ adalah attribute biaya (cost)} \end{cases} \dots\dots\dots(1)$$

Keterangan :

r_{ij} = nilai rating kinerja ternormalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

$\max x_{ij}$ = nilai terbesar dari setiap kriteria

$\min x_{ij}$ = nilai terkecil dari setiap kriteria

benefit = jika nilai terbesar adalah terbaik

cost = jika nilai terkecil adalah terbaik

1. Kriteria Usia

$$r_{11} = \frac{\min \{0.67, 0.67, 1.0, 0.67\}}{0.67} = \frac{0.67}{0.67} = 1$$

$$r_{21} = \frac{\min \{0.67, 0.67, 1.0, 0.67\}}{0.67} = \frac{0.67}{0.67} = 1$$

$$r_{31} = \frac{\min \{0.67, 0.67, 1.0, 0.67\}}{1.0} = \frac{0.67}{1.0} = 0.67$$

$$r_{41} = \frac{\min \{0.67, 0.67, 1.0, 0.67\}}{0.67} = \frac{0.67}{0.67} = 1$$

2. Kriteria Pendidikan

$$r_{12} = \frac{0.25}{\max\{0.25, 0.5, 0.5, 0.5\}} = \frac{0.25}{0.5} = 0.5$$

$$r_{22} = \frac{0.5}{\max\{0.25, 0.5, 0.5, 0.5\}} = \frac{0.5}{0.5} = 1$$

$$r_{32} = \frac{0.5}{\max\{0.25, 0.5, 0.5, 0.5\}} = \frac{0.5}{0.5} = 1$$

$$r_{42} = \frac{0.5}{\max\{0.25, 0.5, 0.5, 0.5\}} = \frac{0.5}{0.5} = 1$$

3. Kriteria Akreditasi

$$r_{13} = \frac{0.67}{\max\{0.67, 0.33, 1, 0.67\}} = \frac{0.67}{1} = 0.67$$

$$r_{23} = \frac{0.33}{\max\{0.67, 0.33, 1, 0.67\}} = \frac{0.33}{1} = 0.33$$

$$r_{33} = \frac{1}{\max\{0.67, 0.33, 1, 0.67\}} = \frac{1}{1} = 1$$

$$r_{43} = \frac{0.67}{\max\{0.67, 0.33, 1, 0.67\}} = \frac{0.67}{1} = 0.67$$

4. Kriteria IPK

$$r_{14} = \frac{0.2}{\max\{0.2, 0, 1, 0.4\}} = \frac{0.2}{1} = 0.2$$

$$r_{24} = \frac{0}{\max\{0.2, 0, 1, 0.4\}} = \frac{0}{1} = 0$$

$$r_{34} = \frac{1}{\max\{0.2, 0, 1, 0.4\}} = \frac{1}{1} = 1$$

$$r_{44} = \frac{0.4}{\max\{0.2, 0, 1, 0.4\}} = \frac{0.4}{1} = 0.4$$

5. Kriteria Pengalaman

$$r_{15} = \frac{0,57}{\max\{0,57, 0, 0,86, 0,14\}} = \frac{0,57}{0,86} = 0,66$$

$$r_{25} = \frac{0}{\max\{0,57, 0, 0,86, 0,14\}} = \frac{0}{0,86} = 0$$

$$r_{35} = \frac{0,86}{\max\{0,57, 0, 0,86, 0,14\}} = \frac{0,86}{0,86} = 1$$

$$r_{45} = \frac{0,14}{\max\{0,57, 0, 0,86, 0,14\}} = \frac{0,14}{0,86} = 0,17$$

6. Kriteria Tinggi Badan

$$r_{16} = \frac{0,8}{\max\{0,8, 1, 0,6, 0,8\}} = \frac{0,8}{1} = 0,8$$

$$r_{26} = \frac{1}{\max\{0,8, 1, 0,6, 0,8\}} = \frac{1}{1} = 1$$

$$r_{36} = \frac{0,6}{\max\{0,8, 1, 0,6, 0,8\}} = \frac{0,6}{1} = 0,6$$

$$r_{46} = \frac{0,8}{\max\{0,8, 1, 0,6, 0,8\}} = \frac{0,8}{1} = 0,8$$

7. Kriteria Berat Badan

$$r_{17} = \frac{\min\{1, 1, 0,4, 1\}}{1} = \frac{0,4}{1} = 0,4$$

$$r_{17} = \frac{\min\{1, 1, 0,4, 1\}}{1} = \frac{0,4}{1} = 0,4$$

$$r_{17} = \frac{\min\{1, 1, 0,4, 1\}}{0,4} = \frac{0,4}{0,4} = 1$$

$$r_{17} = \frac{\min\{1, 1, 0,4, 1\}}{1} = \frac{0,4}{1} = 0,4$$

8. Kriteria Status Pernikahan

$$r_{18} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{28} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{38} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{48} = \frac{\min \{1, 1, 1, 0\}}{0} = 0 = 0$$

9. Kriteria Kondisi Mata

$$r_{19} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{29} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{39} = \frac{\min \{1, 1, 1, 0\}}{1} = 0 = 0$$

$$r_{49} = \frac{\min \{1, 1, 1, 0\}}{0} = 0 = 0$$

Dari hasil perhitungan tersebut maka diperoleh matriks normalisasi (R).

Pada aplikasi, data matriks normalisasi (R) dapat ditunjukkan pada Gambar 4.75 sebagai berikut :

Tabel 4.27 Data Matriks Normalisasi (R)

Alternatif	Kriteria								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
A1	1	0.5	0.67	0.2	0.66	0.8	0.4	0	0
A2	1	1	0.33	0	0	1	0.4	0	0
A3	0.67	1	1	1	1	0.6	1	0	0
A4	1	1	0.67	0.4	0.17	0.8	0.4	0	0

Email Pelamar	Nama Pelamar	Nilai Matriks Normalisasi Kriteria Seleksi (R)								
		Usia	Pendidikan	Akreditasi	IPK	Pengalaman	Tinggi	Berat	Status	Kacamata
arya@gmail.com	Arya Prabowo	1.00	0.50	0.67	0.20	0.66	0.80	0.40	0.00	0.00
Dono@gmail.com	Dono Iswanto	1.00	1.00	0.33	0.00	0.00	1.00	0.40	0.00	0.00
edo@gmail.com	Edo Tambun	0.67	1.00	1.00	1.00	1.00	0.60	1.00	0.00	0.00
jodi@gmail.com	Jodi Setiawan	1.00	1.00	0.67	0.40	0.17	0.80	0.40	0.00	0.00

Gambar 4.75 Uji Coba Menampilkan Data Matriks Normalisasi (R)

Langkah terakhir yang dilakukan untuk memperoleh perankingan adalah dengan penjumlahan dari perkalian matriks ternormalisasi (R) dengan bobot variabel kepentingan (W) setelah dikonversi ke bilangan *fuzzy* sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A) untuk contoh kasus penentuan alternatif pencari kerja untuk jabatan pekerjaan *Dekstop Programmer* pada PT. Mekanika.

- $$V1 = \{(1)(1)+(1)(0.5)+(1)(0.67)+(1)(0.2)+(1)(0.66)+(1)(0.8)+(1)(0.4) \\ + (1)(0)+(1)(0)\} = \\ 1 + 0.5 + 0.67 + 0.2 + 0.66 + 0.8 + 0.4 + 0 + 0 = 4.23$$
- $$V2 = \{(1)(1)+(1)(1)+(1)(0.33)+(1)(0)+(1)(0)+(1)(1)+(1)(0.4) \\ + (1)(0)+(1)(0)\} =$$

$$1 + 1 + 0.33 + 0 + 0 + 1 + 0.4 + 0 + 0 = 3.73$$

$$3. \quad V_3 = \{(1)(0.67)+(1)(1)+(1)(1)+(1)(1)+(1)(1)+(1)(0.6)+(1)(1)+(1)(0)+(1)(0)\} = 0.67 + 1 + 1 + 1 + 1 + 0.6 + 1 + 0 + 0 = 6.27$$

$$4. \quad V_4 = \{(1)(1)+(1)(1)+(1)(0.67)+(1)(0.4)+(1)(0.17)+(1)(0.8)+(1)(0.4)+(1)(0)+(1)(0)\} = 1 + 1 + 0.67 + 0.4 + 0.17 + 0.8 + 0.4 + 0 + 0 = 4.44$$

Nilai terbesar ada pada V_3 sehingga alternatif A_3 atau Edo adalah alternatif yang terpilih sebagai alternatif terbaik. Pada aplikasi, nilai terbesar yang dipilih sebagai alternatif terbaik (A) dapat ditunjukkan pada Gambar 4.76 sebagai berikut:

Peringkat	Nama Kandidat	Keterangan	Skor	Aksi
1	Edo Tambun Email : edo@gmail.com	Usia : 24 Tahun Pendidikan : SI Akreditasi : A IPK / NEM : 4 Pengalaman : 6 Tahun Berkacamata : Tidak Status : Belum Menikah Tinggi Badan : 165 Cm Berat Badan : 67 Kg	6.27	Terima Compare
2	Jodi Setiawan Email : jodi@gmail.com	Usia : 25 Tahun Pendidikan : SI Akreditasi : B IPK / NEM : 3.25 Pengalaman : 1 Tahun Berkacamata : Tidak Status : Belum Menikah Tinggi Badan : 168 Cm Berat Badan : 55 Kg	4.44	Terima Compare

Gambar 4.76 Uji Coba Menampilkan Data Rekomendasi Pencari Kerja

4.3.2 Evaluasi Aplikasi

Evaluasi aplikasi dilakukan dengan penyebaran angket untuk mengetahui bagaimana respon pengguna terhadap sistem yang dibangun. Respon pengguna nantinya dapat dijadikan acuan untuk menentukan kemampuan sistem secara

keseluruhan. Berdasarkan angket yang telah disebar, maka didapatkan data hasil angket sebagai berikut :

Tabel 4.28 Hasil Perhitungan Angket Pencari Kerja

No	Pernyataan	Jumlah Skor
Proses yang dikembangkan pada sistem		
1	Sistem mampu menyajikan informasi lowongan kerja dengan data yang akurat dan mudah dipahami	34
2	Sistem mampu melakukan pencatatan data pencari kerja secara <i>online</i> dengan baik dan benar	32
3	Sistem dapat memfasilitasi pengguna untuk dapat mengecek status lamaran secara <i>online</i> .	30
4	Sistem dapat memfasilitasi pengguna untuk memperbaharui profil kapan saja dan di jaga langsung oleh pengguna.	33
Output yang dihasilkan oleh sistem		
5	Sistem mampu memberikan <i>output</i> histori lamaran dengan data yang akurat dan mudah dipahami	34
Pendapat user akan kemampuan aplikasi, apakah dibutuhkan ?		
6	Kebutuhan untuk menggunakan sistem informasi ini sangat tinggi	34
Kualitas tampilan aplikasi		
7	Desain tampilan mudah dimengerti	35
8	Desain tampilan mudah dipelajari	35
9	Tata letak tampilan aplikasi mudah dikenali	32
10	Teks/tulisan pada layar dapat dibaca dengan jelas	36
11	Desain tampilan konsisten	27
12	Penggunaan layar efisien	27
13	Desain tampilan informatif	33
Pendapat user akan kemampuan aplikasi, apakah sangat membantu?		
14	Menurut saya aplikasi sangat membantu pencari kerja untuk mencari dan melamar kerja	33
Pendapat user mengenai output yang dihasilkan		
15	Saya puas dengan <i>output</i> yang dihasilkan oleh sistem	34

Tabel 4.29 Hasil Perhitungan Angket Penyedia Kerja

No	Pernyataan	Jumlah Skor
Proses yang dikembangkan pada sistem		
1	Sistem ini dapat membantu memenuhi kebutuhan untuk mendapatkan basis data pencari kerja yang akurat dan tercatat dengan baik	33
2	Sistem mampu melakukan pencatatan data lowongan kerja secara <i>online</i> dengan baik dan benar	33
3	Sistem ini dapat memberikan gambaran tentang kemampuan dan potensi yang dimiliki seorang pencari kerja	30
4	Sistem dapat membantu proses seleksi dan <i>filtering</i> dari basis data pencari kerja dengan mudah dan sesuai kriteria yang dibutuhkan.	30
Proses yang dikembangkan pada sistem		
5	Sistem ini dapat meminimalkan kesalahan penilaian terhadap seorang pencari kerja.	28
6	Sistem ini dapat mendukung perusahaan dalam proses pengambilan keputusan rekrutmen dan seleksi calon karyawan	31
Output yang dihasilkan oleh sistem		
7	Sistem mampu menyajikan laporan transaksi, seperti laporan pencari kerja masuk, laporan pencari kerja (per periode dan per posisi), laporan lowongan kerja (per periode, per posisi dan paling diminati) dengan data yang akurat dan mudah dipahami	34
Pendapat user akan kemampuan aplikasi, apakah dibutuhkan ?		
8	Kebutuhan untuk menggunakan sistem informasi ini sangat tinggi	37
Kualitas tampilan aplikasi		
9	Desain tampilan mudah dimengerti	34
10	Desain tampilan mudah dipelajari	34
11	Tata letak tampilan aplikasi mudah dikenali	34
12	Teks/tulisan pada layar dapat dibaca dengan jelas	36
13	Desain tampilan konsisten	28
14	Penggunaan layar efisien	26
15	Desain tampilan informatif	26
Pendapat user akan kemampuan aplikasi, apakah sangat membantu ?		
16	Menurut saya aplikasi sudah memudahkan anda dalam memasukkan informasi dan mendapatkan informasi yang dibutuhkan	28
Pendapat user mengenai output yang dihasilkan		
17	Saya puas dengan <i>output</i> yang dihasilkan oleh sistem	30

Angket menggunakan skor penilaian sebagai berikut :

SS : Sangat Setuju (Skor 4)

S : Setuju (Skor 3)

KS : Kurang Setuju (Skor 2)

TS : Tidak Setuju (Skor 1)

Angket dibagikan kepada dua puluh responden, responden yang dimaksud antara lain sepuluh departemen/divisi yang menjadi wakil dan pihak yang bertanggung jawab dalam melakukan proses rekrutmen dan seleksi dalam perusahaan atau instansi terkait, serta sepuluh para pencari kerja untuk melakukan pengujian aplikasi. Skor minimal untuk setiap pertanyaan adalah sepuluh. Sedangkan skor maksimal untuk setiap pertanyaan adalah empat puluh. Nilai tengah yang didapatkan adalah $(10 + 40) / 2 = 25$. Apabila skor suatu pernyataan kurang dari dua puluh lima, maka dapat dikatakan pernyataan tersebut tidak sesuai atau tidak benar. Sebaliknya apabila skor suatu pernyataan lebih dari dua puluh lima, maka dapat dikatakan pernyataan tersebut sesuai atau benar. Dengan mengacu pada aturan diatas, maka didapatkan kesimpulan untuk setiap pernyataan sebagai berikut :

Tabel 4.30 Kesimpulan Setiap Pernyataan Angket Pencari Kerja

No	Pernyataan	Jumlah Skor	Hasil
Proses yang dikembangkan pada sistem			
1	Sistem mampu menyajikan informasi lowongan kerja dengan data yang akurat dan mudah dipahami	34	Sesuai
2	Sistem mampu melakukan pencatatan data pencari kerja secara <i>online</i> dengan baik dan benar	32	Sesuai
3	Sistem dapat memfasilitasi pengguna untuk dapat mengecek status lamaran secara <i>online</i> .	30	Sesuai
4	Sistem dapat memfasilitasi pengguna untuk memperbaharui profil kapan saja dan di jaga langsung oleh pengguna.	33	Sesuai

Tabel 4.30 Kesimpulan Setiap Pernyataan Angket Pencari Kerja (Lanjutan)

No	Pernyataan	Jumlah Skor	Hasil
Output yang dihasilkan oleh sistem			
5	Sistem mampu memberikan <i>output</i> histori lamaran dengan data yang akurat dan mudah dipahami	34	Sesuai
Pendapat user akan kemampuan aplikasi, apakah dibutuhkan?			
6	Kebutuhan untuk menggunakan sistem informasi ini sangat tinggi	34	Sesuai
Kualitas tampilan aplikasi			
7	Desain tampilan mudah dimengerti	35	Sesuai
8	Desain tampilan mudah dipelajari	35	Sesuai
9	Tata letak tampilan aplikasi mudah dikenali	32	Sesuai
10	Teks/tulisan pada layar dapat dibaca dengan jelas	36	Sesuai
11	Desain tampilan konsisten	27	Sesuai
12	Penggunaan layar efisien	27	Sesuai
13	Desain tampilan informatif	33	Sesuai
Pendapat user akan kemampuan aplikasi, apakah sangat membantu ?			
14	Menurut saya aplikasi sangat membantu pencari kerja untuk mencari dan melamar pekerjaan	33	Sesuai
Pendapat user mengenai output yang dihasilkan			
15	Saya puas dengan <i>output</i> yang dihasilkan oleh sistem	34	Sesuai

Tabel 4.31 Kesimpulan Setiap Pernyataan Angket Penyedia Kerja

No	Pernyataan	Jumlah Skor	Hasil
Proses yang dikembangkan pada sistem			
1	Sistem ini dapat membantu memenuhi kebutuhan untuk mendapatkan basis data pencari kerja yang akurat dan tercatat dengan baik	33	Sesuai
2	Sistem mampu melakukan pencatatan data lowongan kerja secara <i>online</i> dengan baik dan benar	33	Sesuai
3	Sistem ini dapat memberikan gambaran tentang kemampuan dan potensi yang dimiliki seorang pencari kerja	30	Sesuai
4	Sistem dapat membantu proses seleksi dan <i>filtering</i> dari basis data pencari kerja dengan mudah dan sesuai kriteria yang dibutuhkan.	30	Sesuai

Tabel 4.31 Kesimpulan Setiap Pernyataan Angket Penyedia Kerja (Lanjutan)

No	Pernyataan	Jumlah Skor	Hasil
5	Sistem ini dapat meminimalkan kesalahan penilaian terhadap seorang pencari kerja.	28	Sesuai
6	Sistem ini dapat mendukung perusahaan dalam proses pengambilan keputusan rekrutmen dan seleksi calon karyawan	31	Sesuai
Output yang dihasilkan oleh sistem			
7	Sistem mampu menyajikan laporan transaksi, seperti laporan pencari kerja masuk, laporan pencari kerja (per periode dan per posisi), laporan lowongan kerja (per periode, per posisi dan paling diminati) dengan data yang akurat dan mudah dipahami	34	Sesuai
Pendapat user akan kemampuan aplikasi, apakah dibutuhkan?			
8	Kebutuhan untuk menggunakan sistem informasi ini sangat tinggi	37	Sesuai
9	Desain tampilan mudah dimengerti	34	Sesuai
10	Desain tampilan mudah dipelajari	34	Sesuai
11	Tata letak tampilan aplikasi mudah dikenali	34	Sesuai
12	Teks/tulisan pada layar dapat dibaca dengan jelas	36	Sesuai
13	Desain tampilan konsisten	28	Sesuai
14	Penggunaan layar efisien	26	Sesuai
15	Desain tampilan informatif	26	Sesuai
Pendapat user akan kemampuan aplikasi, apakah sangat membantu ?			
16	Menurut saya aplikasi sudah memudahkan anda dalam memasukkan informasi dan mendapatkan informasi yang dibutuhkan	28	Sesuai
Pendapat user mengenai output yang dihasilkan			
17	Saya puas dengan <i>output</i> yang dihasilkan oleh sistem	30	Sesuai

Jadi berdasarkan kesimpulan dari setiap pertanyaan angket seperti pada Tabel 4.29 dan Tabel 4.30, maka didapatkan hasil tiga puluh empat pernyataan dinyatakan sesuai atau benar. maka dapat diambil kesimpulan bahwa kemampuan sistem yang dibangun secara keseluruhan dapat sesuai dengan kebutuhan pengguna.